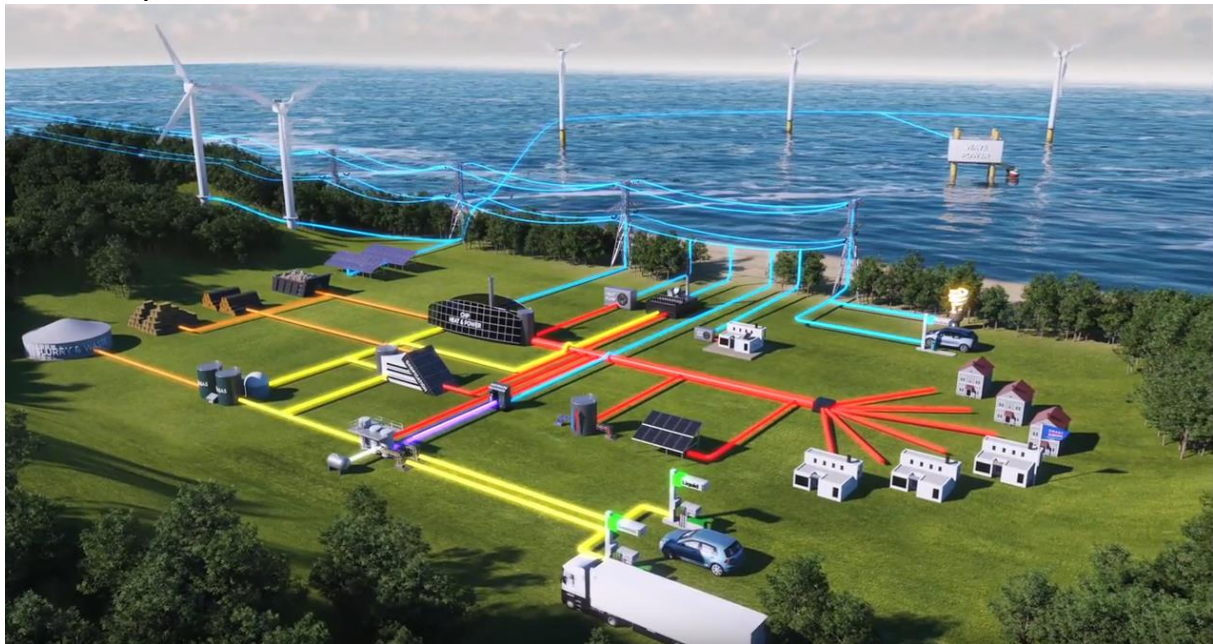


# Smart Solutions: Smart Grid Demokit

## Final Report



30 Jun. 19

Saxion Enschede  
Academie Life Science, Engineering & Design  
Postbus 70.000  
7500 KB Enschede  
(053) 4871111  
[www.saxion.nl](http://www.saxion.nl)

Client Saxion	:	VAN LEEUWEN, RP (RICHARD)	
	:	SCHAEFER, E (EDMUND)	
Coach Saxion	:	JANSSEN, F (FRITS)	
Student + student numbers	:	RIJKEBOER, MF (MAAIKE)	444562
	:	DIJKSTRA, M (MARTIJN)	438476
	:	MARKVOORT, J (JUSTIN)	435804
	:	SONCK, D (DANIËL)	457011
	:	SCHOLTEN, VS (VINCENT)	438528
	:	LÓPEZ, N (NÚRIA)	473880
	:	BRINKHUIS, T (THIJS)	455539
Group	:	Smart Grid Demokit (L.24676)	

## Contact page

---

Client: Dr. Ir. Richard P. van Leeuwen  
r.p.vanleeuwen@saxion.nl  
+31 (0) 610892455  
W3.53

Client: Edmund Schaefer  
e.w.schaefer@saxion.nl  
0880191380

Coach: Frits Janssen  
f.janssen@saxion.nl  
0880196086  
W3.39

Student: Maaïke F. Rijkeboer  
444562@student.saxion.nl  
+ 31 (0) 640020939

Student: Martijn Dijkstra  
438476@student.saxion.nl  
+ 31 (0) 640232872

Student: Justin Markvoort  
435804@student.saxion.nl  
+31 (0) 648699322

Student: Daniël Sonck  
457011@student.saxion.nl  
+31 (0) 683554997

Student: Vincent S. Scholten  
438528@student.saxion.nl  
+31 (0) 658950247

Student: Núria López  
473880@student.saxion.nl  
+34 (0) 675520879

Student: Thijs Brinkhuis  
455539@student.saxion.nl  
+31 (0) 620788550

## Preface

---

The report shown here concludes the project called "Smart Grid Demokit" written by Maaïke Rijkeboer, Martijn Dijkstra, Justin Markvoort, Daniël Sonck, Vincent Scholten, Núria López and Thijs Brinkhuis. Our group contains an international student from Spain. Our backgrounds are Mechanical Engineering, Industrial Product Development, Applied Computer Science and Electrical Engineering.

This report is produced in the period from 1 February 2019 to 1 September 2019, in which we created a complete demokit.

This project was commissioned by the lectorate Sustainable Energy Supply and the University of Twente.

We want to thank Richard P. van Leeuwen and Edmund Schaefer for the good communication as our client and giving us the opportunity to do this project. Also, we like to thank Gerwin Hoogsteen for the introduction to Demokit helping us to work with it. Last, we like to thank Frits Janssen for the accompaniment as our tutor in this project. He really helped us work as a group.

We hope you enjoy reading our report!

Maaïke Rijkeboer, Martijn Dijkstra, Justin Markvoort, Daniël Sonck, Vincent Scholten, Núria López and Thijs Brinkhuis

July 2019  
Enschede

## Summary

---

The purpose of this report is to justify the design choices of the smart grid demo kit. Something had to be designed to make a smart grid clear for people who have little knowledge about smart grids. The product had to be appealing and clear for people to understand. And eventually should be usable, for example, on an information market.

The first part of the research consisted of looking how to shape the whole system. How the 'tiles' had to look to be interactive for users and what they should feature. One part of this was doing research to get to know more about the already existing knowledge amount users. Another research investigated what appeals the most to the users.

After this, a concept was created in compliance with the group and the client. The concept consists of hexagonal tiles, each with a different function: houses, solar panels, wind turbines, factories and energy storages. These tiles are all different parts of a smart grid. When combining these tiles, it can be made clear to users how smart grids work. The tiles are fabricated using a combination of 3D printing and laser cutting. The tiles have laser cut symbols on top of them to show what part of the smart grid they are. Digital LED strips are on top of the tiles to show the direction of the energy flow, and the colors indicate if the tile is producing or consuming power from the grid.

The tiles are connected to each other by the so called “grid blocks”. These blocks make up the central power grid and are also lighting up by LED strips. Each tile is equipped with a microcontroller which controls the LED strips and makes it possible for the different tiles to “talk” with each other. Using this, the central tile knows which tiles are connected to the system. The central tile controls all tiles and runs the simulation of the smart grid.

For further development of the project, it can be investigated how to control and adjust the system from an external system, for example by a tablet. The final product consists of five tiles connected by seven grid blocks which show how a smart grid works.

## List of figures

Figure 1: Morphological overview of the designs .....	6
Figure 2: Concept 1 design .....	7
Figure 3: Concept 2 design .....	7
Figure 4: Concept 3 design .....	8
Figure 5: Concept 4 design .....	9
Figure 6: Magnet concept design .....	11
Figure 7: Horizontal slide design .....	12
Figure 8: Vertical sliding design .....	12
Figure 9: Horizontal sliding slot design .....	13
Figure 10: Nuts and bolts design .....	13
Figure 11: Male female header connectors .....	16
Figure 12: Overview of the connections on the tiles .....	19
Figure 13: Connections on the grid block .....	19
Figure 14: Circuit of the tile board .....	20
Figure 15: PCB design .....	22
Figure 16: Solar Power Tile .....	27
Figure 17: Animation of the solar tiles .....	27
Figure 18: Energy storage Tile .....	28
Figure 19: Battery charge states .....	28
Figure 20: Battery discharging .....	29
Figure 21: Industry Tile .....	29
Figure 22: Energy flow of Industry Tile .....	29
Figure 23: Houses Tile .....	30
Figure 24: Energy flow of the houses tile from left to right .....	30
Figure 25: Wind Power Tile .....	31
Figure 26: Energy flow of Wind power tile from left to right .....	31
Figure 27: All tiles connected together .....	32
Figure 28: Architecture overview .....	33
Figure 29: Timing diagram of the attention line .....	35
Figure 30: Packet construction .....	36
Figure 31: Side with space for diffuser .....	39
Figure 32: Side without space for diffuser .....	39
Figure 33: Connection block with led strip on top .....	39
Figure 34: Connection block with led strip inside .....	39
Figure 35: Bottom side of the connection block .....	40
Figure 36: Inlay male USB .....	40
Figure 37: Inlay female USB .....	40
Figure 38: Different possibilities for led diffuser .....	40
Figure 39: Printing the sides of the tiles .....	41
Figure 40: The USB's during printing .....	41
Figure 41: The USB's during printing .....	41
Figure 42: The led diffuser during printing (top) The led diffuser with led (bottom) .....	41
Figure 43: Laser engraving and cutting off the tiles .....	42
Figure 44: The assembly process of a tile .....	42
Figure 45: Printing a connection block .....	42
Figure 46: Connection block and cable layout .....	43
Figure 47: A connection block from the bottom (left), and from the top (right) .....	43
Figure 48: Process of creating a PCB .....	44
Figure 49: The produced PCBs .....	44
Figure 50: Finished board .....	45

Figure 51: Male USB connector block .....	45
Figure 52: Running LED test program.....	48
Figure 53: Response to interrupt line.....	49

## Table of Contents

---

Contact page .....	ii
Preface .....	iii
Summary .....	iv
List of figures.....	v
Table of Contents .....	vii
1. Introduction .....	1
2. Research model.....	2
3. System Requirements .....	5
4. High level design.....	6
4.1 Design choices.....	6
5. Low level design .....	11
5.1 Mechanical.....	<b>Fout! Bladwijzer niet gedefinieerd.</b>
5.2 Electrical.....	15
5.3 Visualization.....	24
5.4 Software.....	33
6. Realisation .....	39
6.1 Drawings of the components .....	39
6.2 Production of the tiles.....	41
6.3 Software.....	46
7. Results and testing .....	48
7.1 Software.....	51
8. Discussion .....	Fout! Bladwijzer niet gedefinieerd.
9. Conclusion.....	52
10. Recommendations .....	53
11. Reflection .....	Fout! Bladwijzer niet gedefinieerd.
12. References .....	Fout! Bladwijzer niet gedefinieerd.
13. Appendix .....	Fout! Bladwijzer niet gedefinieerd.
14. Literature.....	Fout! Bladwijzer niet gedefinieerd.
15. Appendix .....	56
15.1 Planning and scheduling .....	56
15.2 List of requirements.....	57
15.3 2D-Drawings.....	60

# 1. Introduction

---

More than 12 billion of euros per year could be saved in the energy bills of citizens by 2022 thanks to the Smart Grids (Smith, 2018). A research shows that Smart Grids development in cooperation with the Smart Cities, will allow citizens saving that amount of money because of installing smart energy meters (Smith, 2018).

In 2000, the “Telegestore” project was born in Italy being the first grid that used 29.8 million smart meters connected to control the energy in Italy (IEEE ISPLC , 2007). At present, many countries and institutions around the world have decided to invest and implement the use of smart grids for their capacity to dealing with global warming (Munuera, 2019).

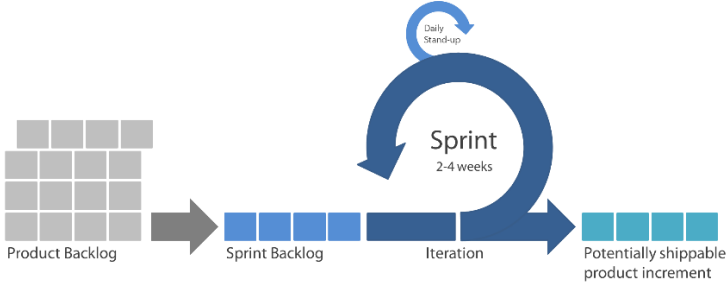
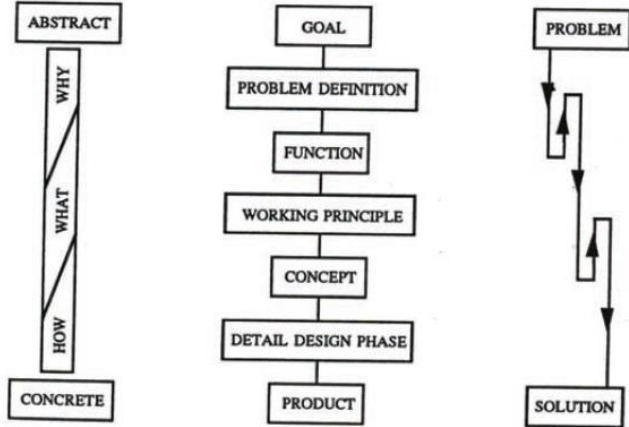
Smart grid is a network that integrates the information technology to the existing electricity grid to optimize energy efficiency through the bidirectional exchange of electrical information, in real time, between suppliers and consumers (Smart grids, 2019). Smart grids harmonize the needs and capabilities of all generators, network operators and end-users to make an efficient use of the system. In addition, the costs are minimized and the impact environment as well, while maximizing reliability, strength and stability of the system (International Energy Agency, 2013). Generally, electricity prices increase along with demand. By providing consumers with information about current consumption and energy prices, smart grid energy management services help to minimize the consumption during high-cost, peak-demand times. Moreover, this technology includes the decentralization of power generation.

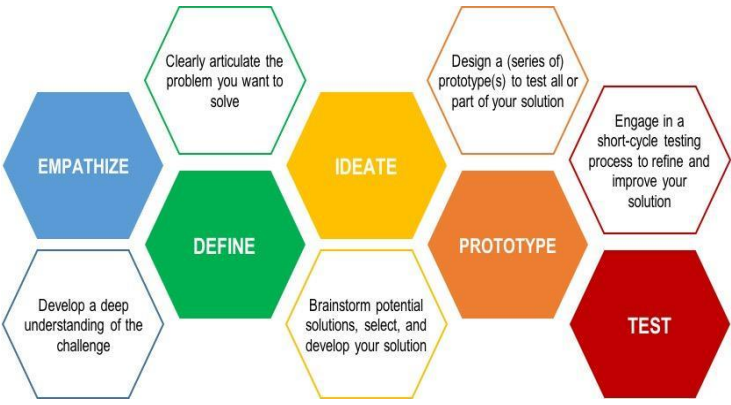
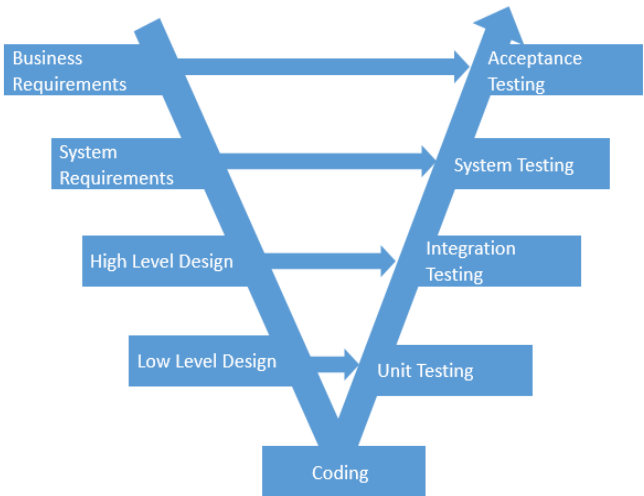
However, the knowledge about Smarts Grids are not so present in the society despite contributing a long list of benefits not only for the economy of people but also for avoiding waste of energy produced. Therefore, the main goal in this project is educate people who do not know how a Smart Grid works creating a demo kit which can simulate and visualize how Smart Grids work. The demokit will be composed of different tiles and each tile will represent elements of a Smart Grid; for example, houses, solar power, energy storage or industry.



## 2. Research model

Any design process or methodology can perfectly lead to all situations that require solutions in a project. Specially, when a work team is made up of students from different fields. Next, an overview is given of the studies of the team and their corresponding methodologies or design processes.

<p>Scrum</p>	<p>Scrum is an Agile methodology working technique which, through collaborative practices, all types of risks are minimized in the preparation of a project. This method was created in high productivity teams.</p> <p>In Scrum there is no final project delivery, but partial deliveries are made on a regular basis. This is what gives most benefits to the client of the project. For this reason, Scrum is especially suitable for complex environments, where changes occur very frequently and where speed, flexibility, adaptability and competence play a fundamental role.</p>	 <p>The diagram illustrates the Scrum process. It begins with a 'Product Backlog' represented by a stack of grey squares. An arrow points to the 'Sprint Backlog', shown as a row of blue squares. From the Sprint Backlog, an arrow leads into a 'Sprint' cycle, which is a large blue circle labeled 'Sprint 2-4 weeks'. Inside the cycle, a smaller blue circle indicates a 'Daily Stand-up'. The cycle ends with an arrow pointing to 'Potentially shippable product increment', represented by a row of teal squares.</p>
<p>Kroonenberg</p>	<p>Kroonenberg created a design process as a chain of activities, which starts with an abstract problem and its results in a concrete solution. The design process is divided in three stages: the definition of the problem, the selection of the working principle and the detailed design. Dividing the design process into stages generates easier tasks to the way to solve the problem.</p>	 <p>The diagram shows the Kroonenberg design process as a vertical flowchart. On the left, a vertical bar is divided into three sections labeled 'WHY', 'WHAT', and 'HOW', with 'ABSTRACT' at the top and 'CONCRETE' at the bottom. The main flowchart consists of a series of boxes connected by downward arrows: 'GOAL', 'PROBLEM DEFINITION', 'FUNCTION', 'WORKING PRINCIPLE', 'CONCEPT', 'DETAIL DESIGN PHASE', and 'PRODUCT'. To the right, a separate flow shows 'PROBLEM' leading to 'SOLUTION' through a series of iterative steps, represented by arrows that loop back to previous stages.</p>

<p>Design Thinking</p>	<p>The Design Thinking is a method to generate innovative ideas that focuses its effectiveness on understanding and solving the real needs of users. The Design Thinking process consists of five stages. It is not linear. At any time, you can go backwards or forwards if you see fit, jumping even to non-consecutive stages.</p>	 <pre> graph TD     EMPATHIZE[EMPATHIZE Develop a deep understanding of the challenge] --&gt; DEFINE[DEFINE Clearly articulate the problem you want to solve]     DEFINE --&gt; IDEATE[IDEATE Brainstorm potential solutions, select, and develop your solution]     IDEATE --&gt; PROTOTYPE[PROTOTYPE Design a series of prototype(s) to test all or part of your solution]     PROTOTYPE --&gt; TEST[TEST Engage in a short-cycle testing process to refine and improve your solution]     TEST --&gt; EMPATHIZE   </pre>
<p>V-model</p>	<p>It is a procedure that provides a guide for planning and carrying out projects. It consists of a sequence of steps which all the activities that must be followed for the development of the project are described. Method V means "verification and validation". With this methodology, the characteristics of the product are continuously tested (validation) and then, the step made is validated if it fulfils the purpose for which it was created. Each step in the project is tested and validated.</p>	 <pre> graph TD     BR[Business Requirements] --&gt; ST[System Testing]     SR[System Requirements] --&gt; IT[Integration Testing]     HLD[High Level Design] --&gt; UT[Unit Testing]     LLD[Low Level Design] --&gt; C[Coding]     C --&gt; UT     UT --&gt; IT     IT --&gt; ST     ST --&gt; AT[Acceptance Testing]   </pre>

	Pros	Cons
<b>Scrum</b>	Delivering completed and the most prioritized requirements every month or every two weeks provides a lot of advantages. For example, regular management of customer expectations and based on tangible results, anticipated results (time to market). Flexibility and adaptation to customer needs like changes in the market. Systematic risk mitigation of the project. Alignment between the client and the development team.	This method would be the most useful for the client thanks to the solutions deliveries in every sprint, but it also can present some risks. This team is composed by students from different fields so the tasks in the project are not completely defined from the beginning. This could unbalance the time in the sprint deadline. Moreover, the team is not experienced in this methodology, only two of seven people that compose the team have worked with this scrum before. That increase the risk of losing the timetable programmed.
<b>Kroonenberg</b>	The design process is divided into stages to generate easier tasks. This eases the way to the final delivery.	This design process can be useful but the basic problem with this method is the lack of information in English and the mechanical engineers' students of the team are not so familiar with it.
<b>Design Thinking</b>	The user to whom the project is directed is analysed very well by techniques of market research and user analysis. The projects are previously tested by the user to get their feedback, so the chances of success and cost savings are much higher.	It must be practised before to pass effortlessly through every stage. Specially in empathize step because most of the team members are from technical fields. Sometimes Design Thinking methodology must be accompanied by other methodologies to complete all the fields of a project. In addition, the steps time are not defined so it can be an endless process.
<b>V-model</b>	This methodology specifies well the different roles for the tasks. Checks must be done in every waterfall model stage, so mistakes can be localised fast. It is more robust and complete model than Waterfall model and software can be produced with better quality. Moreover, the problem complexity is solved as soon as possible and not through iterations and development cycles.	The client must be patient because the product will be delivered at the end of the process. Every test can delay the deadline and it can cost a loss of money if it is not effective. Finally, the final product cannot gather all the client requirements.

In this project a physical product must be designed. This product is a combination of different fields that must complement each other and the probability of failure at the final delivery without test the steps done is high. Therefore, is important to use a testing model to guarantee the correct combination of the mechanical, electrical and software elements. V-model is the design process which can warrant the check and validation in every important step and localising possible mistakes.

### 3. System Requirements

---

There are quite a few system requirements for the Smart Grid Demokit, the complete list of requirements is listed in Appendix 11.2 below. The most important requirements are listed below here.

For the functional requirements the Demokit should have different tiles, and each tile will represent another part of the smart grid. At the end of this project there will be at least 4 tiles representing houses, solar/wind park, factory and storage. Another functional requirement is that the tiles should be connected fast with each other (within 10 seconds). There should be a visual representation of the energy flow. And the entire Demokit should be possible to transport without damaging the tiles.

For the technical requirements, the tiles should be greater than 400 \* 400 mm to prevent the that the total assembly will be too thick.

## 4. High level design

### 4.1 Design choices

The design question of the tiles consists of a few different pieces, the main design can be divided into looks and functionality. For the looks the opinion of the client is the most important, the functionality is predefined in the list of requirements, but how these different functions are fulfilled is something that must be thought out.

#### 4.1.1 Looks of the demo-kit

To define how the demo-kit should look, a few possible designs were thought up. Each concept was graded based on factors that were defined together with the client. First the different concepts will be listed and explained.

To get an overview of all the possible combinations a morphological chart with all the different components was made. Because we use the V-model the morphological chart is only used as an overview and the detailed choices are not drawn in the chart.


















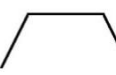






Function		Methods					
Nr.	Description	1	2	3	4	5	6
A	Design tiles	 Octagon	 Rectangular	 Square	 No center tile		
B	Mechanical connection	 Nut bolt	 Slot	 Hook + magnet	 Snaphooks	 Lego	
C	Electrical connection	 USB	 Improvise	 Wireless	 AUX	 Pogo pins	
D	Shape for tile	 Square	 Triangular	 Rectangle	 Trapezoid	 Rounded trapezoid	 Hexagon
E	Visualization of the energy flow	 LED to the tiles	 LED around tiles	 Pixel grid	 Motor controlled arrow		
F	Game design	Knobs/buttons to change/turn off tiles	Add costs as a factor	Stable system			

Figure 1: Morphological overview of the designs

### Concept 1 (octagonal center tile with smaller tiles all around)

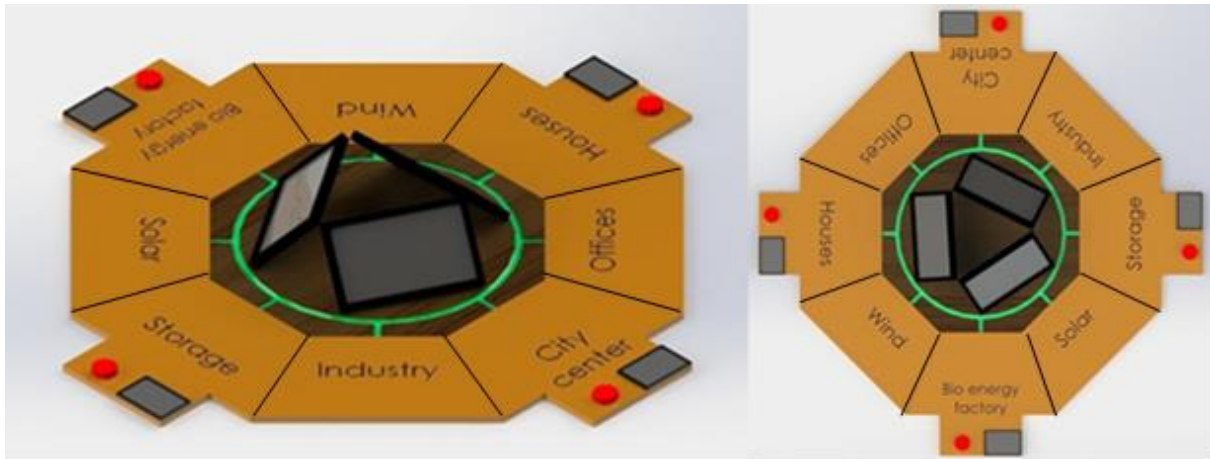


Figure 2: Concept 1 design

The center tile functions as the heart of the whole system to which all different tiles can be connected (Up to a maximum of eight tiles). The smaller tiles can be connected to the eight sides of the octagon. All these tiles communicate to the center tile which is the master tile. The center tile runs the simulation, shows the information for the user and communicates with the eight smaller tiles. The smaller tiles all contain standard hardware which makes it possible to connect buttons, potentiometers, screens and motors. This hardware also controls the visualization of the energy flow for that tile. To transport the system the different small tiles can be disconnected from the main tile. The smaller tiles can be put in a box and can be transported this way.

RGB LED strips are used on both the center tile as on the smaller tiles to visualize the energy flow. The strips on the smaller tiles show the energy flowing in and out the tile and the strips on the center tile represent the grid which can be different colors according to the state of the system.

### Concept 2 (rectangular center tile with smaller tiles on either side)

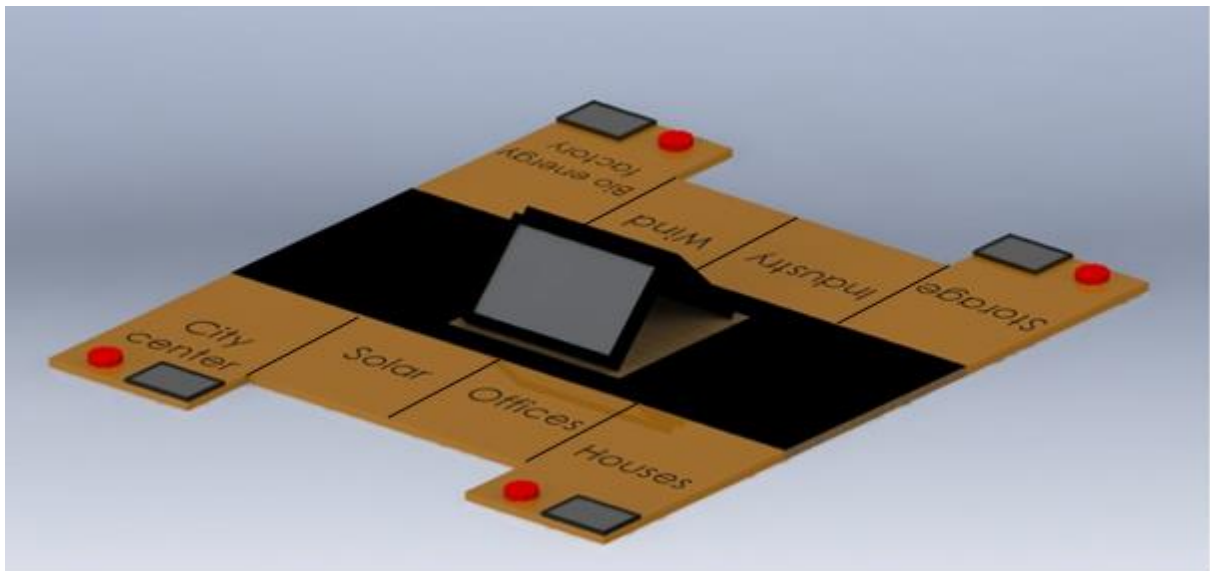
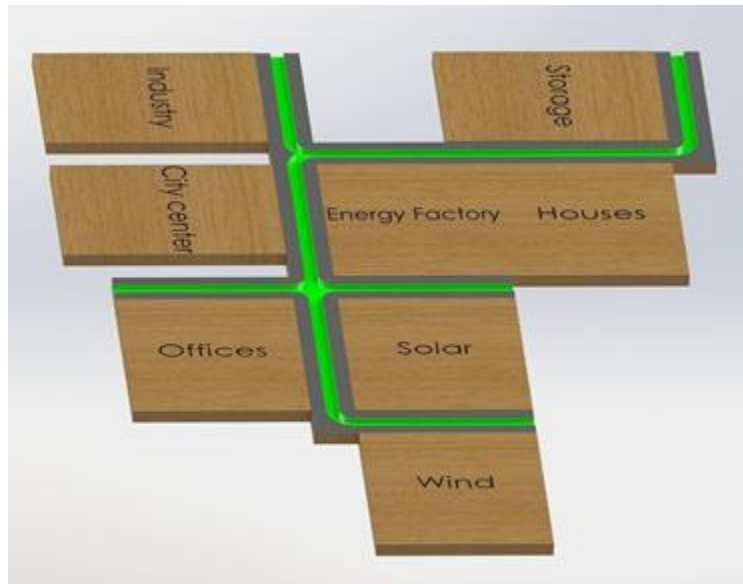


Figure 3: Concept 2 design

This design consists of a rectangular center tile where square tiles can be connected to all four sides. All these tiles communicate to the center tile which is the master tile. This tile runs the simulation, shows the information for the user on screens and communicates with the eight smaller

tiles. The top of the center tile is covered in a pixel grid which is used to show the energy flow in the grid, this pixel grid can also be used to show additional information to the user. The system can be expanded by just making the center tile longer. The smaller tiles all contain standard hardware which makes it possible to connect buttons, potentiometers, screens and motors. This hardware also controls the visualization of the energy flow for that particular tile.

### **Concept 3(Rectangular tiles with parts of the grid connected)**



*Figure 4: Concept 3 design*

All the tiles can be connected side to side to another. On the front of each tile there is a part of the grid. To go around corners or make junctions, special smaller tiles can be used. Therefore, when the tiles are connected, they form the central grid where the energy flows. There is no maximum of tiles that can be connected because there are no position limitations. This means new tiles simulating new scenarios can be produced and added to the demokit. The tiles can be positioned as the user wants, simulating streets. The tiles also contain standard hardware which makes it possible to connect buttons and potentiometers but there are no screens on the demo-kit. Therefore, to display information, a simple monitor can be connected to show additional information about the system.

#### Concept 4 (tiles which sides form the central grid)

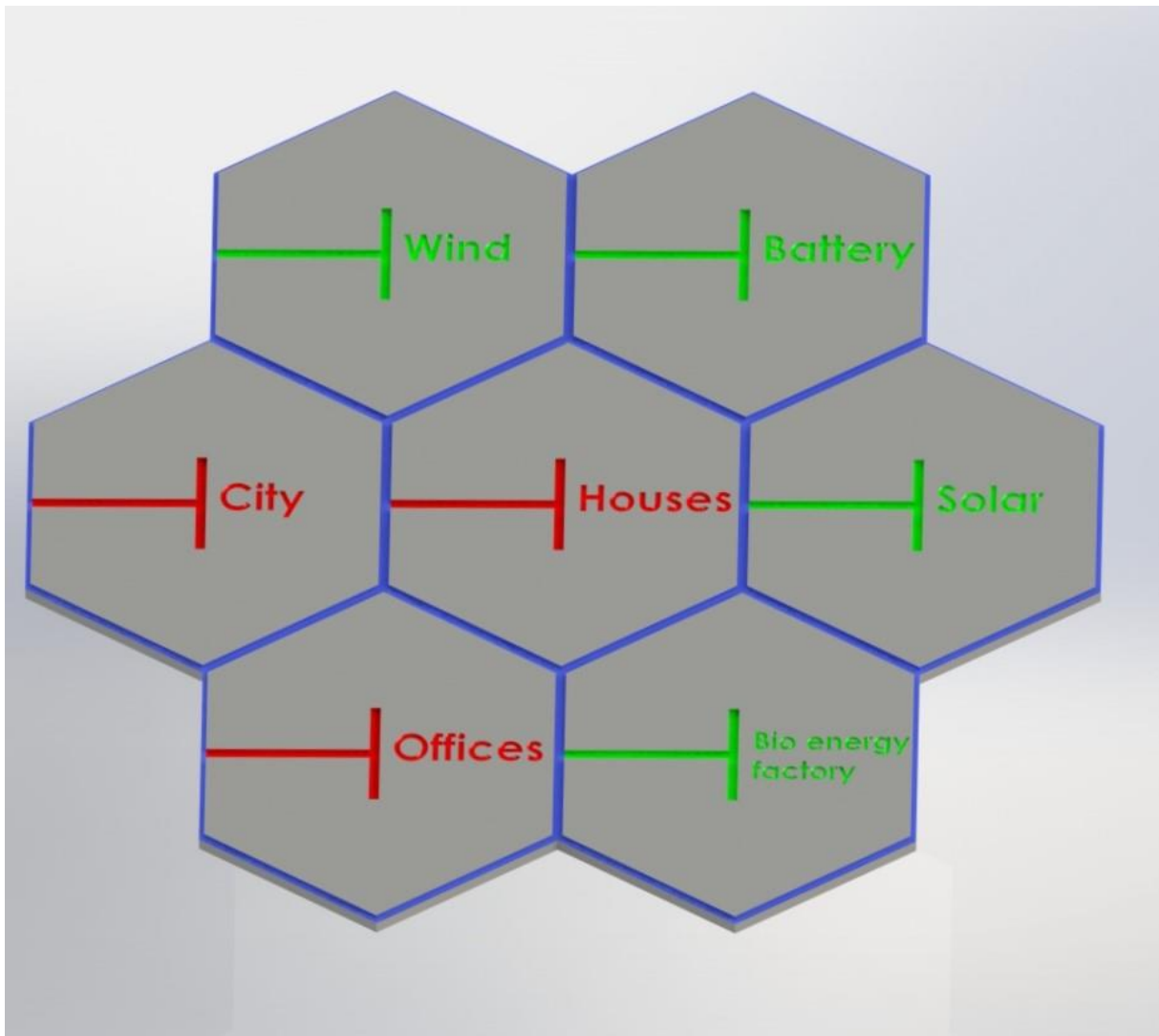


Figure 5: Concept 4 design

This concept consists of hexagonal tiles which represent parts of a smart grid. The concept contains one master tile which controls all different tiles. The tiles have a connection on each of the six sides, making it possible to connect the tiles in all ways possible. The sides of the tiles are filled with RGB LED's and when tiles are connected to each other, these sides make the central grid of the system. The advantage of this system is that it is not limited to a maximum number of tiles. Next to the LEDs on the side of the tiles, it also has a strip on the tile itself showing the energy flowing into or out of the tile. Just as the other concepts, the tiles can contain buttons, potentiometers, screens or motors. This is done to make the system more interactive with the user.



### 4.1.2 Scoring for the concepts

The concepts will be graded using the criteria that were composed together with the client. The criteria are listed and explained below, these main criteria also contained several sub criteria to make it easier to give a well-supported score.

1. Logical shape  
The product should look visually attractive and it must be self-explaining.
2. Expandability  
It should be possible to extend the system with new tiles, this will cause software problem and hardware difficulties.
3. Possibility to show energy flows.  
The demo-kit must be able to show energy flows, this can be done in multiple ways, the more ways that are possible the better.
4. Producibility  
The easier a system can be produced the better the end product will be.
5. Transportability  
The system must be transportable, when most of the components are shaped similar and when the components aren't too big this is easier.

Each criterion had its one weight factor that also had to be filled in together with the score for the criteria. All the criteria were put into an excel document and sent to the client to fill in, as a group the document was also filled in and together with the results of the client this gave the following score for the concepts:

	Concept 1	Concept 2	Concept 3	Concept 4
Logical shape	82,8	88,4	72,0	86,7
Expandability	42,3	43,1	53,7	60,6
Possible energy flow representation	64,2	77,4	59,0	64,3
Producibility	8,8	9,6	10,6	11,9
Transportability	33,1	35,4	42,0	50,5
<b>Total</b>	<b>231</b>	<b>254</b>	<b>237</b>	<b>274</b>

Table 1: Scoring of the concepts

Concept 4 scored the best in almost every category. Overall it scored the best by far and that's why this concept was chosen to work out in detail. In the Appendix 11.3 the whole overview of the total scores can be found.

## 5. Low level design

---

### 5.1 Mechanical connections

For the mechanical connections five possibilities were thought up. They will all be explained and in the end a conclusion will be drawn in which the choice will be explained.

#### 5.1.1 Concepts

##### Magnets (With the possibility of integrated electrical connections)

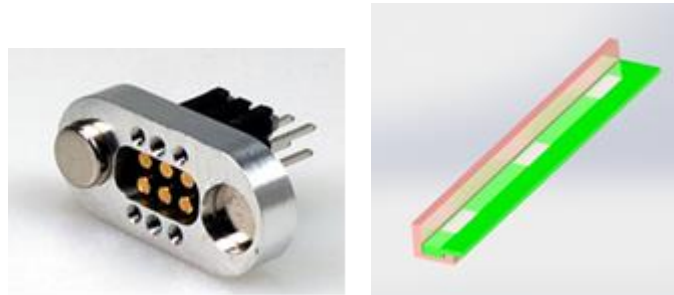


Figure 6: Magnet concept design

Magnets can be used for the mechanical and electrical, the advantage of magnets for the mechanical connections is that the connection can be made to be self-aligning. By making use of the north and south poles of magnets they can be made to find each other and assure a proper connection. Also, the conductivity is a useful feature because this makes it possible to implement the electrical connections.

A downside of using magnets is that the magnets lose magnetism after a couple years of use.

Pros	Cons
Mechanical and electrical connection in one	Magnets lose attraction power over time
Self-aligning	Relative expensive

## Horizontal sliding connections



Figure 7: Horizontal slide design

Pins on one side of the connector will be slid into pockets on the other side, one of the tiles is then moved in the horizontal plane to lock into place. The mechanical part of this connections is very straight forward and doesn't need any other materials besides the material that is needed for the connector itself (no magnets for example). A downside of this concept is that the implementation of the electrical connections is relatively difficult because the contact surfaces slide past each other which makes it difficult to assure a proper connection. Another downside is that this makes it difficult/impossible to use all the sides of a hexagonal shaped tile because the sliding movement that is limited to the horizontal plane.

Pros	Cons
Can be 3d printed at ones (no inlays)	Electrical connection is difficult to implement
Strong solid mechanical connection	Sliding in the last tile in a hexagonal shape is impossible

## Vertical sliding connections

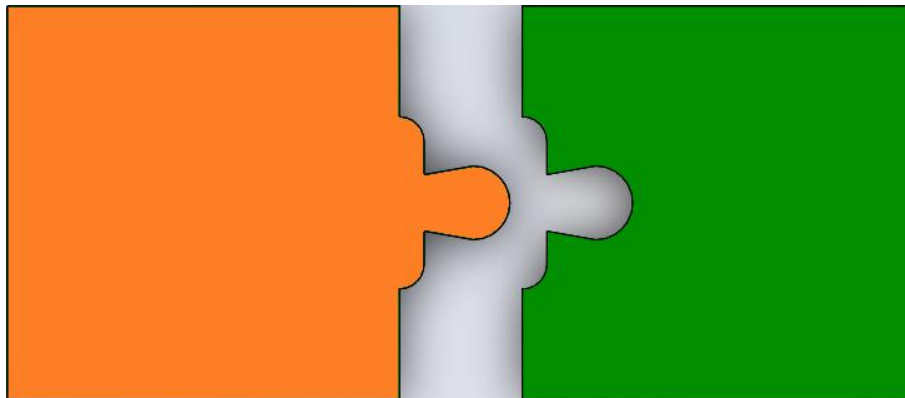


Figure 8: Vertical sliding design

This concept is based on a puzzle. The pieces are connected into place from the top. This ensures a sturdy connection because there won't be a lot of movement in the vertical plane when the demo-kit is in use. This type of connection also allows a lot of different types of electrical connection to be used because there is a surface on which the tile is pressed to another tile and there is a surface on which the tiles slide past each other. A downside of this concept can be that the tiles must be thick to make it possible to place the electrical connections in the vertical plane. Another downside is that there are male and female tiles, this limits the number of configurations that are possible.

Pros	Cons
Sliding in the last tile in a hexagonal shape is possible	The concept will be thick because of the electrical connection
Sturdy connection	Male and female connectors are a must what will limit the number of configurations

## Horizontal sliding slot connections

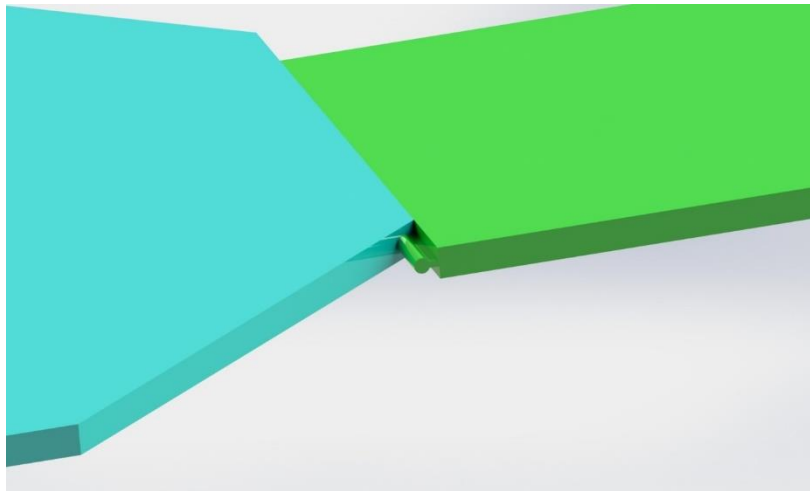


Figure 9: Horizontal sliding slot design

Another way to connect the tiles is by using a slot connection method, this variant a bit the same as the other horizontal sliding connection, the tiles should be shifted in from the side. It is a strong connection but the problem with this connection is that the last tile is impossible to get in when a hexagonal shape is used.

Pros	Cons
Can be 3d printed at ones (no inlays)	Electrical connection is difficult to implement
Strong solid mechanical connection	Sliding in the last tile in a hexagonal shape is impossible

## Nuts and bolts



Figure 10: Nuts and bolts design

This concept uses nuts and bolts to connect the tiles together, this is an easy and fool proof connections type which is easy to make and very strong. A downside of this connections type is that the implementation of the electrical connections will be difficult. Also, this type of connection is relatively slow and might be difficult when a lot of tiles are connected to each other.

Pros	Cons
Very strong	Electrical connection will be connected separately
Easy production	Not fast and easy to assemble on side
Cheap	

### 5.1.2 Concept choice mechanical *connections*

Because the hexagonal shape is chosen, a connection type which allows connections on all the six sides of the tile is necessary. This means there can't be a male or female side on the tiles because then the tiles aren't universal. This is the case for multiple of the connection types, a possibility to alleviate these limitations is to use connection blocks between the tiles. The connection blocks will allow all the sides off the tile to be the same so the whole system can be connected in every way. Using connection blocks will allow all the different connection types as discussed above to be used on all sides off the tiles.

To make the tiles fast to connect, the connection should be made in one movement, this eliminates the nuts and bolts because this will take too long and will also be difficult when a lot of tiles are connected. The horizontal sliding connection limits the number of tiles that can be connected to one tile because there is some space needed to slide it in to place, taking this and the difficulty to implement the electrical components into account also eliminates this concept.

The remaining concepts are both possible and the final choice must be made after the different possibilities for the electrical connections have been reviewed because these two are closely related to each other.

## 5.2 Electrical

### 5.2.1 Electrical connection concepts

Now that the shape of the tiles for the product have been chosen, it is important to choose the right electrical connections between the tiles. The electrical connections are needed for the tiles to communicate with each other. The connections are also used to power up the “single” tiles. There are different aspects which must be considered when choosing the electrical connections. Aspects such as: strength, stability, durability, safety, easy to use and producibility.

The different concepts for the electrical connections are stated below, they will be rated according to the aspects mentioned above.

#### POGO pins

POGO pins are male connectors which have a spring inside of them. These male pins connect to a pad which is the female side of the connection. This spring makes sure that the pin itself has a tight connection against the pad it must be connected with. When there is a little distance between the connectors, the spring will make sure the pin and the pad won't disconnect from each other.

The pogo pin connection must be combined with a mechanical connection, because the pogo pin connection itself cannot keep different tiles tight to each other.

Pros	Cons
Stability	Strength
Durability	Producibility
Easy to use	Costs

The stability is an advantage of this concept because the pogo pins have an internal spring which keeps the connection stable during slight movement of the tiles. The strength is a disadvantage because the pogo pins itself can't keep a mechanical connection between tiles. The pins need a mechanical connection to stay connected with each other. The durability is an advantage of the pogo pins, because pogo pins are strong and can hold for a very long time. The producibility is a disadvantage of this concept because all pins must be placed separately instead of using just one connector. The connection is easy to use, because the tiles just must be connected to each other by the mechanical connection and the pogo pin connection will just follow automatically. The user doesn't have to take the pogo pin connection into account when connecting tiles.

## Male female headers

Male female headers are connectors which can be seen in the figure below.

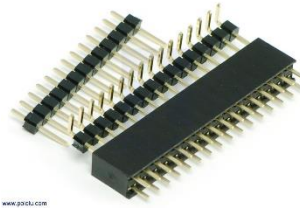


Figure 11: Male female header connectors

The pins are all connected to each other by a plastic strip, making the spacing between the pins equal.

Pros	Cons
Stability	Strength
Durability	Easy to use
Producibility	
Costs	

The stability of this concept is an advantage, because when the pins are connected to each other, they will stay connected to each other. The strength is a disadvantage in this design, because the pins are weak and can easily bend or even break when the connection is put under stress. The connection is not easy to use for users who aren't into electronics, because users must know how the connection works. The durability of this connection is an advantage because it is an easy connection, making it durable when properly installed. The producibility is an advantage in this concept because all pins of a connection are connected to each other by a plastic strip. This makes only one connector to install on each side of the tiles.

## Wireless connection

A wireless connection can be something like Bluetooth or WIFI. The tiles could have receiving and transmitting Bluetooth modules. This way they can send and receive information to each other. The only problem with this is that there is still a wired connection needed to power up the tiles or each tile should have its own power source.

Pros	Cons
Easy to use	Stability
Durability	Producibility
	Costs

Since there is no physical connection, the strength will not be considered. An advantage of this concept is that it is extremely easy to handle for the user since there is no connection and everything happens automatically. However, it is complex to create wireless communication with many tiles at the same time. The stability is also a disadvantage because wireless connection is not always stable. At last there is the disadvantage of the costs, to have Bluetooth modules in all the different tiles will be quite expensive.

## USB connection

A USB connector is a connector which has four pins. The connection consists of a male and a female part. The male part can be connected inside the female connector.

Pros	Cons
Easy to use	Producibility
Stability	
Costs	
Strength	

It is easy to use because it is “plug and play” and everyone knows how a USB connector works. The strength scores very high for this concept since it does not only provide an electrical connection but also a mechanical connection. Durability is a little advantage because it has a relatively long lifetime. USB connectors are stable if they are plugged in there is almost no way that they are not connected properly. At last the producibility and the costs. The USB connectors are not that expensive and therefor the costs are an advantage. The producibility is stated as a disadvantage but this is not a big problem. The only disadvantage is that it will take a lot of time to solder all the wires to the connectors.



### 5.2.2 The chosen concept

After looking at all the advantages and disadvantages of the different electrical connections. The choice has been made to use the USB connectors. They provide strength and stability and besides that they are extremely easy in use.

### 5.2.3 Implementation of the connectors in the tiles

Six different electrical lines are important to make the whole system work together. These will be discussed below:

- **+5V**  
This line, combined with the GND line, will supply the power for the whole system. The +5V will be the supply voltage for the PCBs as well as for the LEDs. The +5V will be connected to all six sides of the tiles.
- **GND**  
The ground line, combined with the +5V, line will make sure the whole system is supplied with power. The ground line will be connected to all sides of the tiles.
- **RX**  
The RX line is the receiving data line for that tile. It makes sure the tiles can communicate with each other using serial communication. The RX line only goes from a side of the tile to the corresponding RX connector on the PCB.
- **TX**  
The TX line is the transmitting data line for that tile. It sends the data from that tile to the next tile. The TX line only goes from the TX connector on the PCB to the corresponding side of the hexagonal tile.
- **INT (Interrupt)**  
The interrupts line is a line between tiles which is kept high using pull-up resistor. When a tile pulls this interrupt line low, it means that this tile wants to communicate with the other tile. The interrupt line only goes from the PCB to the corresponding side of the PCB.
- **Data grid**  
The data grid is the LED data for all grid blocks. All LED strips in the grid blocks are connected to +5V, GND and the grid data. This grid data tells the LED strip what to do. Because the grid data is connected everywhere in parallel, all grid blocks will display the same making it a fully functioning power grid.

The choice was made to use two USB connectors on each side of the tile, making eight connections on each side of the tile. Only six connections are necessary. Because two pins on the USBs are left, they are used for extra +5V and GND. This splits the current over both connectors. The connection on the tile can be seen in the following figure. The tiles are equipped with two male USBs on each side. As can be seen, the +5V, GND and data grid are all parallel connected to each side of the tile. So, the tiles and grid blocks can be connected to all six sides of a tile.

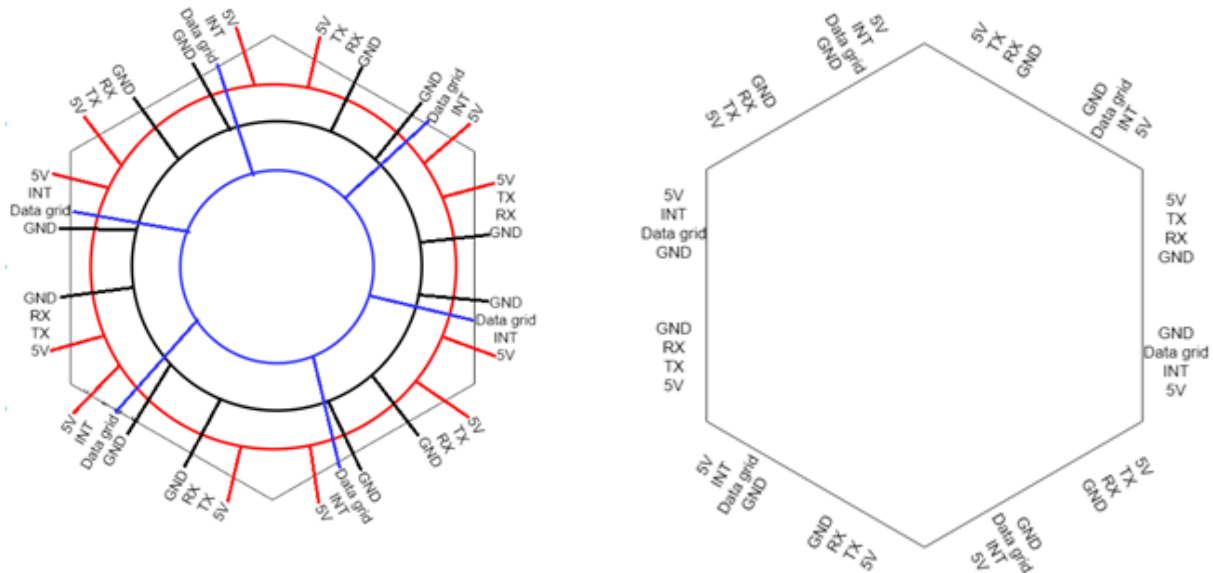


Figure 12: Overview of the connections on the tiles

The grid blocks make the connection between the different tiles possible. The grid blocks have two female USBs on each side, and these connect the tiles to the other side of the grid block. These lines are crossed to make sure two tiles can be connected to each other. The grid blocks are equipped with digital LED strips which use the +5V, GND and the Data grid line. This data grid line sends the data to the LEDs and is parallel connected over the whole system, making all grid blocks having the same LED visualization. Below is a figure of the connections of the grid blocks:

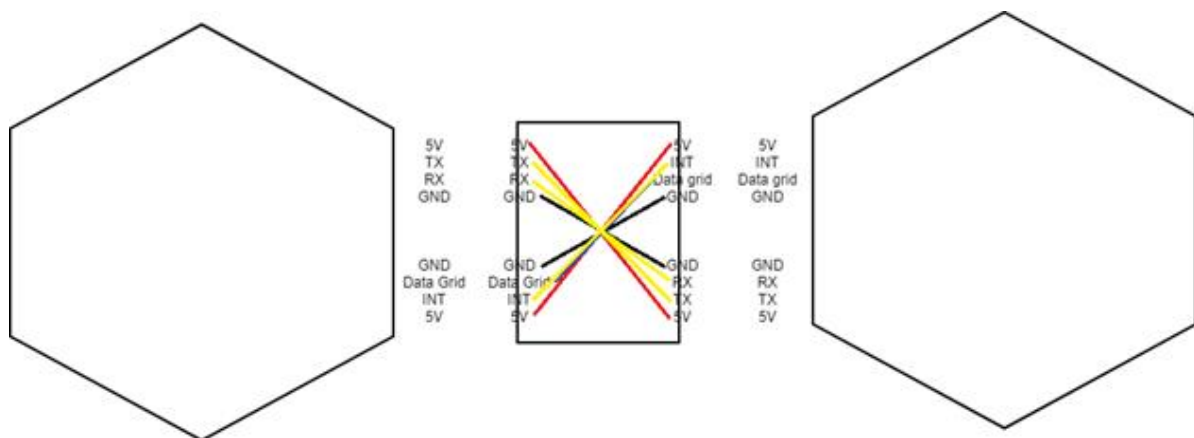


Figure 13: Connections on the grid block

### 5.2.4 Circuit

The figure below shows the circuit with all its different parts. Following will be an explanation of each different part of the circuit.

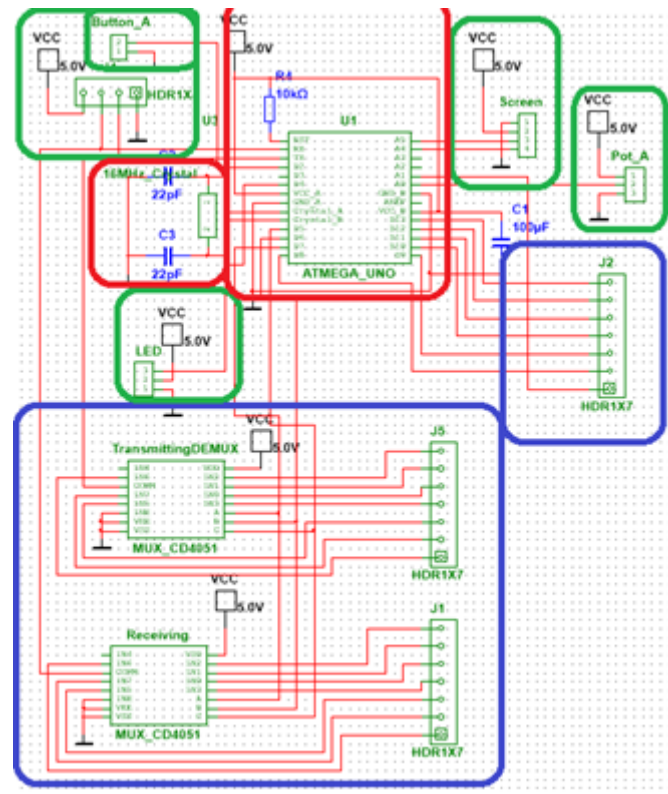


Figure 14: Circuit of the tile board

The circuit can be split up into different parts: The microcontroller, the communication part and all external connections. These will be discussed below.

- **The microcontroller (red)**

The microcontroller is the heart of the circuit because it regulates communication and controls the LED strips. The chosen microcontroller is the ATmega328P, also used in the Arduino Uno boards. This makes it easy to program the chip, because it can be placed in an Arduino Uno board to program the chip using the boards bootloader. The microcontroller is powered by +5V and has a 10kOhm pull-up resistor to disable a reset of the system. A 16MHz crystal is connected to the chip for the clock of the microcontroller. A 100uF capacitor is added to the +5V line of the system to minimize voltage dips in the system.

- **The communication part (blue)**

The communication between the different tiles is made possible by using serial communication, which uses the RX (receiving) and TX (transmitting) ports of the ATmega328P. The problem with this is that all tiles have six sides which all can have different tiles connected to it. All these tiles can't connect to the same RX and TX because the microcontroller does not know which tile it is talking to and communication errors will occur. To solve this problem two (de)multiplexers are used. These are both 8-channel (de)multiplexers which make it possible to switch the communication to all of the six sided of the tile. Now the only problem left is to switch to the right side of the tile to receive data from that particular side. This is done using interrupts. Each side of the tile has a interrupt line connected to the ATmega328P to let the chip know it wants to send data to it. When an

external tile sends an interrupt to the ATmega328P, the system knows which side of the tile to use for communication. And both the RX and TX (de)multiplexer will switch to that side. Both (de)multiplexers have 8 input/output channels which are controlled by 3 bits (A,B and C). These bits are set by the ATmega328P to control which channel is enabled. Both (de)multiplexers are connected to the same bits, making sure that both switch to the right side of the tile. Six channels are used for all the sides of the tiles and the seventh channel is used to communicate to an optional system within the tile. This could be a Raspberry which controls the whole system. The communication part of the system consists of three different parts:

- **Interrupt connector**  
The interrupt header connects each side of the tile directly with the ATmega328P chip. So, if a connected tile wants to communicate to the chip, it will first send an interrupts to the ATmega328P. After this the system knows which side of the tile wants to communicate, so the (de)multiplexers can be switched to the right channel.
- **Transmitting data demultiplexer**  
The transmitting data demultiplexer will take care that the data from the ATmega328P will be send towards the right tile. The used (de)multiplexer is the CD4051 which is an 8-channel (de)multiplexer. After sending an interrupt to the tile it wants to communicate with, the demultiplexer will be set to the corresponding channel and the data transmission can begin.
- **Receiving data multiplexer**  
The receiving data multiplexer will make sure that the incoming data will be coupled with the RX port of the ATmega328P. After an interrupt is received, the chip knows which side of the tile wants to communicate with the system. The receiving data multiplexer will be switch to the corresponding channel and the receiving of data can start.
- **The external connections (green)**  
The circuit has extra external connections to do other thing than only the communication which are listed below:
  - **Communications header (rx and tx)**  
This header makes it possible to communicate with a PC. When connecting it with an Arduino board it can be connected to a computer and being read out. This header also features +5V and a GND to power the whole board and can be connected using jumper wires.
  - **Button connector**  
This connector makes it possible to connect a button or a switch to the system. This button can be placed on top of the tile to make the whole system more interactive. The connector will be a screw terminal making it easy to connect wires to it.
  - **Screen (I<sup>2</sup>C) connector**  
The screen connector adds the feature of placing a screen on top of the tile. The screen can make the tiles more interactive for the user. The connector consists of a +5V, a GND, an SDA and a SCL line. So instead of a screen, also other components using I<sup>2</sup>C can be connected to this connector. This connector will be a screw terminal making it easy to connect an external component to it.
  - **Potentiometer connector**  
The potentiometer connector makes it possible to add a potentiometer to the tile. Instead of a potentiometer also another (analog) component can be added because the connector has an analog port on it next to a +5V and a GND pin. This is also a screw terminal, so it can be connected easily.

- **LED connector**

The LED connector sends the data to the digital LED strip. The digital LED strip has three connections which are also in the LED connector (+5V, GND and Data). The data pin is connected to a digital pin of the ATmega328P which will send the data to the LED strip.

### 5.2.5 The PCB

The PCB design can be seen below and is also divided into different parts. Red for the microcontroller, blue for the communication part and green for the external connections. The PCB is double sided and has a ground plane on both sides. The channel numbers are printed on top of the PCB on the silkscreen for the interrupt, receiving and transmitting connector. All the chips (ATmega328P and the CD4051) are placed on sockets making it possible to remove them from the circuit. The PCB has four holes in it which make it possible to attach it properly into the tile using a screw.

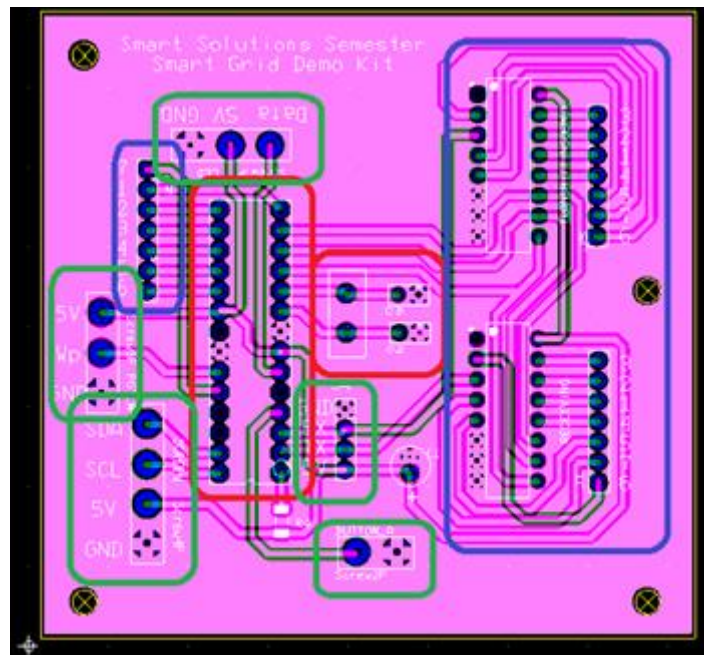


Figure 15: PCB design

Below is the list of each component used for the PCB:

Component name	Amount	Info
ATMEGA 328p	1	28 pins
DIP socket 28 pins	1	Socket for ATMEGA 328p
16MHz crystal	1	Crystal for ATMEGA 328p
22pF capacitors	2	Capacitors voor crystal
Screw terminal 2 pins	1	5mm pin spacing
Screw terminal 3 pins	2	5mm pin spacing
Screw terminal 4 pins	1	5mm pin spacing
CD4051 mux/demux	2	16 pins
DIP socket 16 pins	2	Socket voor CD4051
Capacitors 100uF	1	Power supply capacitors (5V)
Resistor 10k Ohm	1	Pull-up resistor
Header 4pin	1	2.54mm pin spacing
Header 7 pin	3	2.54mm pin spacing

Table 2: Used components

Below is a table which channels of the (de)multiplexer correspond with each side of the tile and which interrupt. The bits of the (de)multiplexer (A, B and C) can be set so the multiplexers match with the right side of the tile.

Tile side	Mux / demux	ABC	Arduino INT pin
1	IO0	000	D8
2	IO1	100	D9
3	IO2	010	D10
4	IO3	110	D11
5	IO5	101	D12
6	IO6	011	D13
7	IO7	111	A1

Table 3: Tile side, (de)multiplexer, interrupt relations

The system will be powered using a special designed power block before mentioned in the mechanical part. This power block has an input connector, a buck converter and 2 female USB connectors. This buck converter generates a voltage of +5V and a maximum of 2 Ampere which powers the whole system. The USB connectors make it possible to connect it to any tile.

## 5.3 Visualization

### User study

When a product must be designed, a lot of factors must to be considered. For example, the costs, materials, mechanical connections, materials resistance, etc. However, one of the most important elements to consider are the users.

At present, any of the products are used daily has been refined so that its use is optimal for people. Something what is decisive is that people understand and know how to use the product because a bad experience of interplay can demolish all the work done.

Therefore, the first step that must be done is studying the users; their interests, skills and needs. Because of that, a survey designed to know these elements was sent to different people. Some questions have been asked to know if people are interested in the topic of the product or if there is any possibility to feel down by the Demo kit, if they know some aspects of Smart Grid and the elements that they could identify properly.

### SKILLS

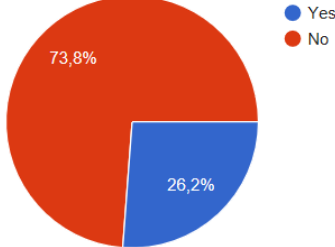
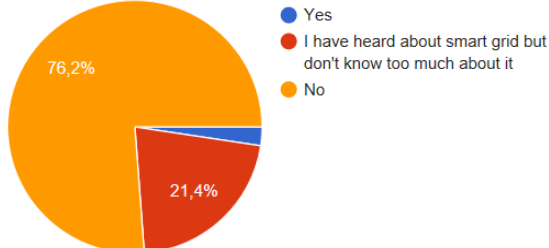
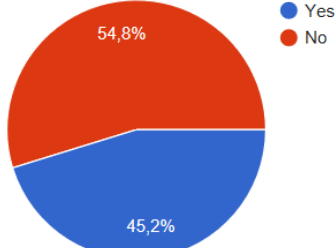
Do you know what electrical consumption peaks are?	 <p>Legend: Yes (blue), No (red)</p> <p>26,2% Yes, 73,8% No</p>	Almost three fourths of the respondents do not know about consumption peaks. It means that people are going to play the smart grid game without previous knowledge about it.
Do you know what a smart grid is?	 <p>Legend: Yes (blue), I have heard about smart grid but don't know too much about it (red), No (yellow)</p> <p>76,2% No, 21,4% I have heard about smart grid but don't know too much about it, Yes (small slice)</p>	it is easy to understand that people are not familiar with the topic. Approximately 98% do not know what a smart grid is. It indicates that the game cannot be too complicated to comprehend.
Do you know what a Smart Home or domotic installation is?	 <p>Legend: Yes (blue), No (red)</p> <p>45,2% Yes, 54,8% No</p>	This answer shows that people have an idea of a centralized system that control

Table 4: Data extracted from the survey

Due to the number of different elements that must be represented in the tiles, various questions had been created to determine what the easiest way is to identify them.



– Solar power

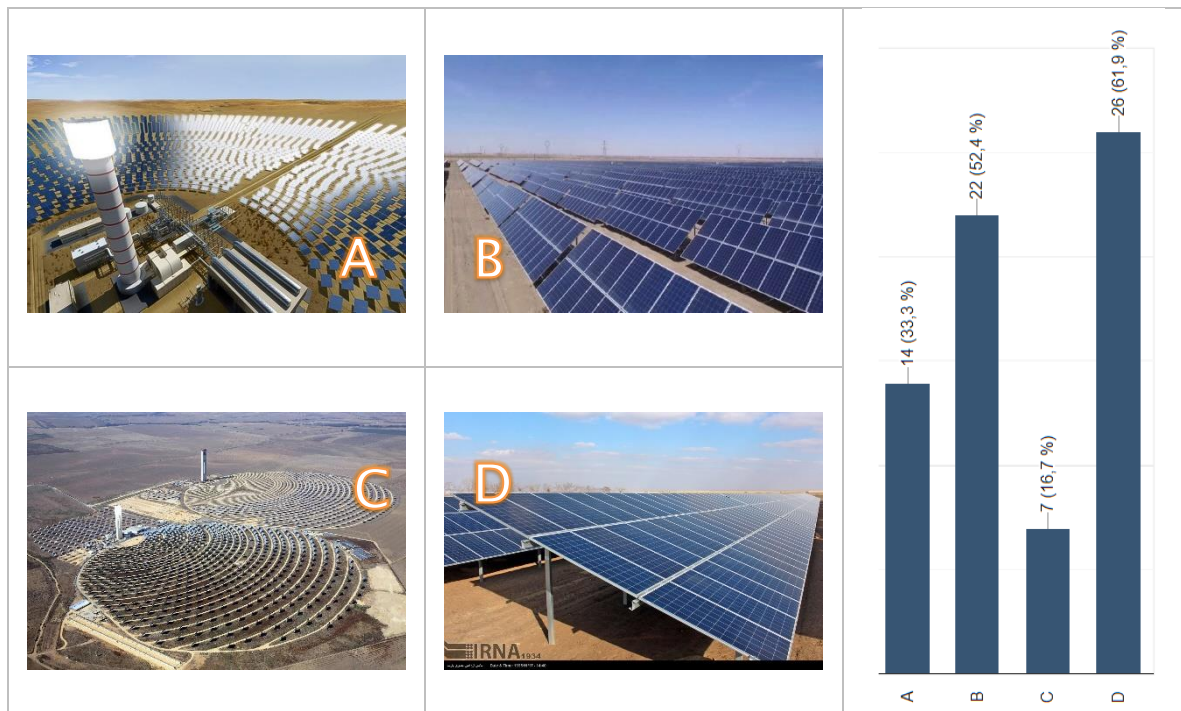


Table 5: Data extracted from the survey

People identify the solar photovoltaic better than the solar power. Consequently, solar power will be represented by photovoltaic panels like in B and D options.

– Energy storage



Table 6: Data extracted from the survey



This question shows how people recognize the energy storage. Clearly, a conventional battery will be used to represent the storage. A detail that could not be forgotten is that option E wins in terms of answers. Therefore, a battery level interface should accompany the battery to help people how much energy is available.

## - Industry

Thanks to the questions about the industry plants, it is known that many people cannot distinguish power production plants from industrial plants that require the energy to obtain a product. It can be concluded that people identify power plants through cooling towers specially like nuclear plants have. Regarding the representation of industrial plants that need big cooling towers cannot appear, at least not in the foreground.

## INTERESTS

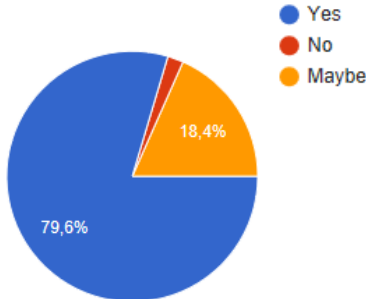
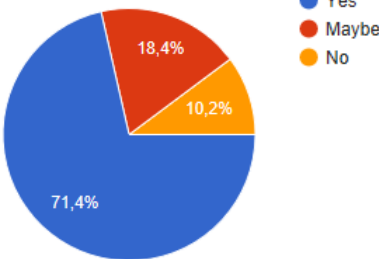
Are you interested in making more efficient and rational use of energy?	 <p>Legend: Yes (blue), No (red), Maybe (orange)</p> <p>79,6% (Yes), 18,4% (Maybe), 2% (No)</p>	Only 2% of interviewees are not interested in making more efficient and rational use of energy. This percentage shows that the great majority could be attracted by the product.
Do you consider that knowing your energy consumption is beneficial?	 <p>Legend: Yes (blue), Maybe (orange), No (red)</p> <p>71,4% (Yes), 18,4% (Maybe), 10,2% (No)</p>	In addition, this theory may be strengthened knowing that only the 10,2% believes that the knowledge of the electrical consumption is not beneficial. The rest of the people are interested in this matter, so the product might be an attraction for them.

Table 7: Data extracted from the survey

## NEEDS

With the survey it has been found that people complain against the amount of money that it would be paid for a smart home installation. It may be that one of the needs of them is not to waste the money. Therefore, people need to be educated showing them the benefits of a smart grid due to it would save us money reducing the waste of energy.

## Design

### Solar power

According to the survey and the research, people identified solar power as a set of solar photovoltaic panels instead of a solar power plant. Therefore, for the representation of this concept a set of six solar panels are drawn with laser cutter.

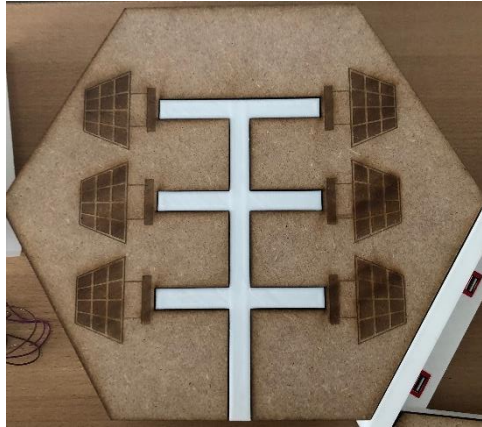


Figure 16: Solar Power Tile

This is one of the tiles that represents energy production and it is going to be represented by the flow of green light that come from each solar panel to the single path.

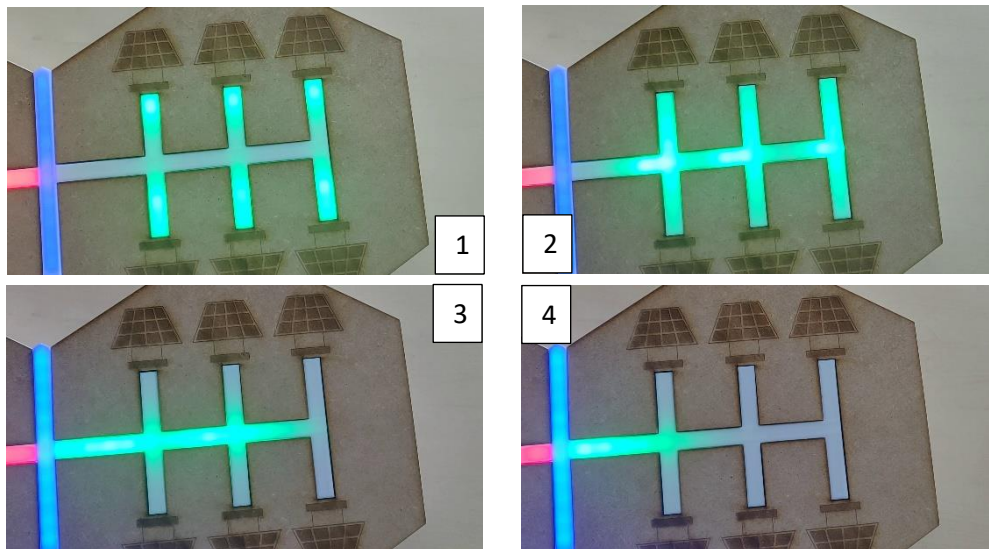


Figure 17: Animation of the solar tiles

## Energy storage

The energy storage is represented by a battery drawing accompany with a battery interface as the results of the survey say.

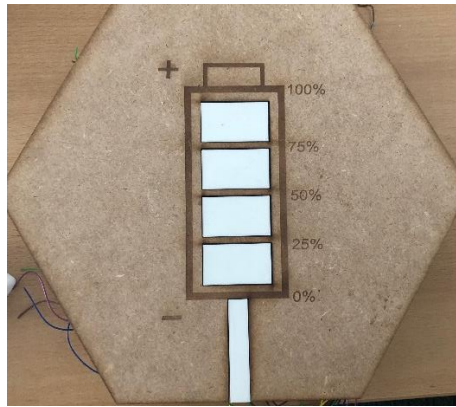


Figure 18: Energy storage Tile

There are five stages for the battery depending on how much energy is stored. When there is no energy, no light flows. If there is a 25% of energy stored, the bottom block is illuminated with red light. When there is 50%, the light of the next block is orange. Finally, when the battery achieves the 75% the light is yellow. And lastly, green with 100%.

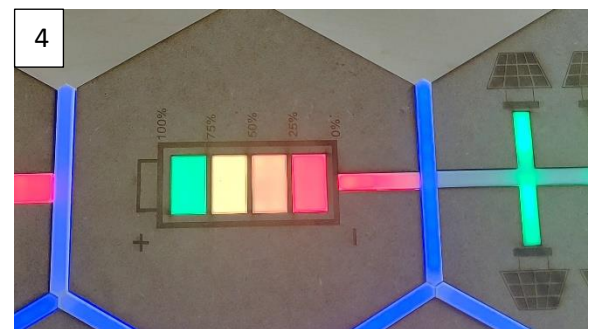
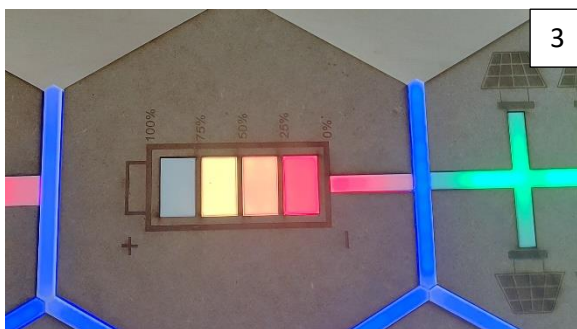
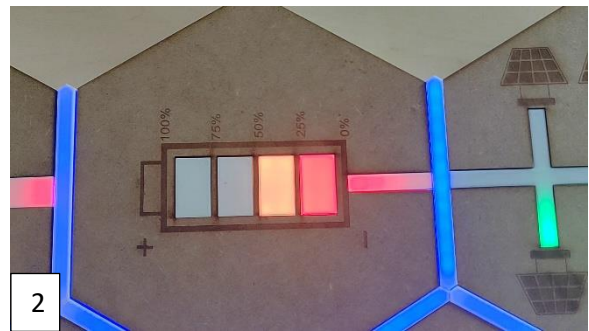
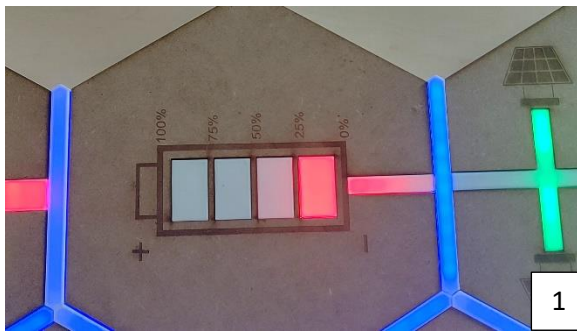


Figure 19: Battery charge states

In addition, the line leading up to the battery will show whether it's consuming/charging by showing a red flow. Or it's producing/discharging by showing a green flow like shown below.

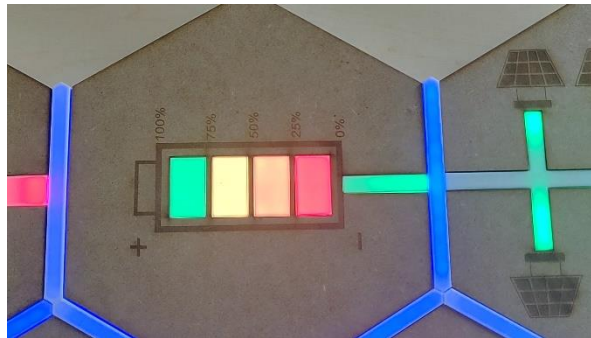


Figure 20: Battery discharging

## Industry

The industry in this Demokit is represented by a factory which consumes a large amount of power. To represent the factory, big cooler towers like nuclear ones cannot be drawn because it could confuse people with power producer plants. However, a building without any tower looked like an ordinary building. Therefore, to make it clear, two towers are pictured in the background.



Figure 21: Industry Tile

This tile depicts big consume so a flow of red light arrives from the grid to the centre of the tile.



Figure 22: Energy flow of Industry Tile



## Houses

Using an identifiable house shape, this tile represents a neighbourhood that consume energy. In real life the houses could produce their own power with solar panels but in this case, the solar panels will be represented in a different tile to make it clearer and easier to understand

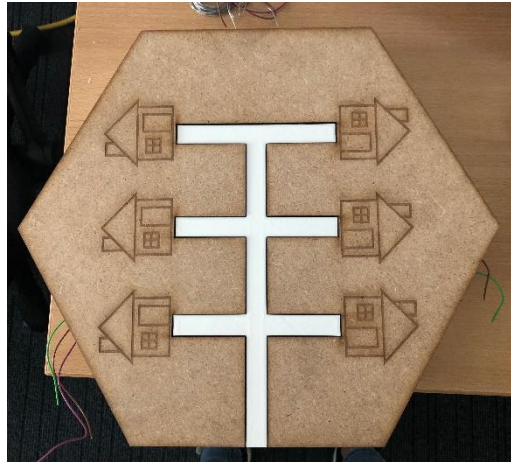


Figure 23: Houses Tile

The flow in this tile will be represented by red color led which flows from the single path (situated in the bottom of the picture) to the bifurcations simulating the electricity arriving to each house from the grid.

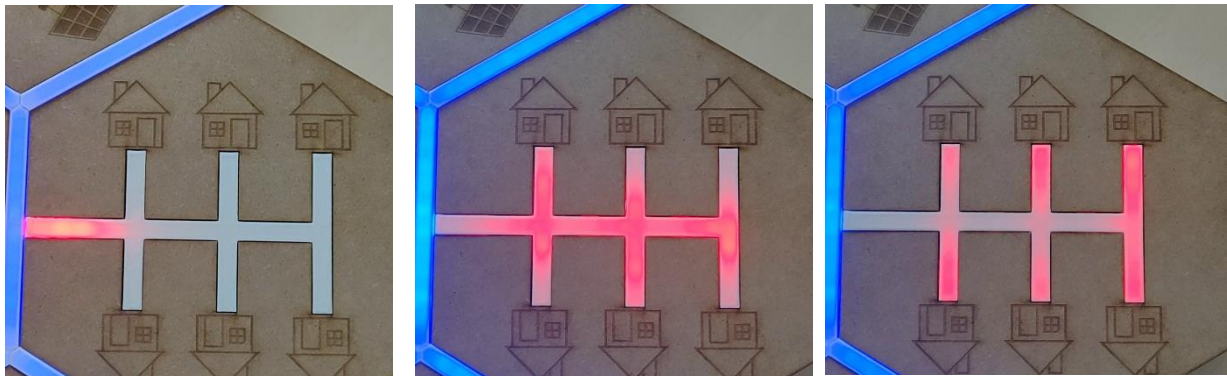


Figure 24: Energy flow of the houses tile from left to right

## Wind power

The wind power is represented by a wind park composed by five wind turbines shapes. For the rendering of this and houses concept a survey was not necessary because of the lack of shape complexity.



Figure 25: Wind Power Tile

In the wind park, energy is produced so the color for the electricity visualization is green. The flow comes from each wind turbine to the single path. It represents that the energy generated goes to the grid.



Figure 26: Energy flow of Wind power tile from left to right

Here is the result connecting the five tiles between each other making a grid represented by the blue light. All the tiles have their own flow of light and they give “energy to the main grid” (tiles with green light) or receive “energy from the grid” (tiles with red light).

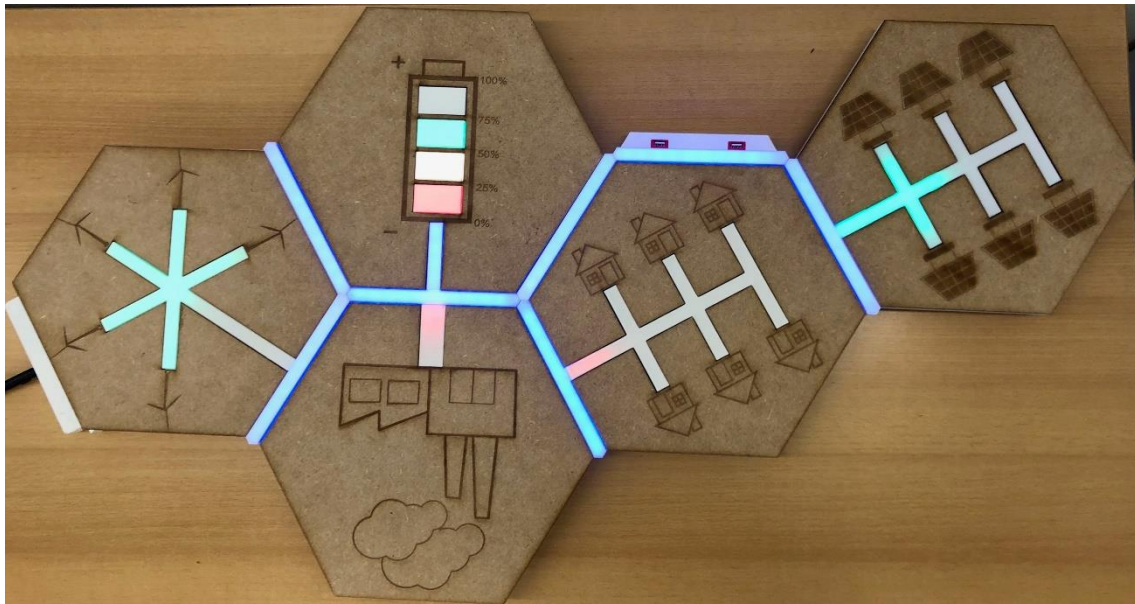


Figure 27: All tiles connected together

## 5.4 Software

The tile and host software are responsible for several different things:

- Control LEDs to visualize energy flow as calculated by Demkit
- Identify which tiles are connected and which types
- Setup Demkit simulation according to the connected tiles
- Run Demkit simulations

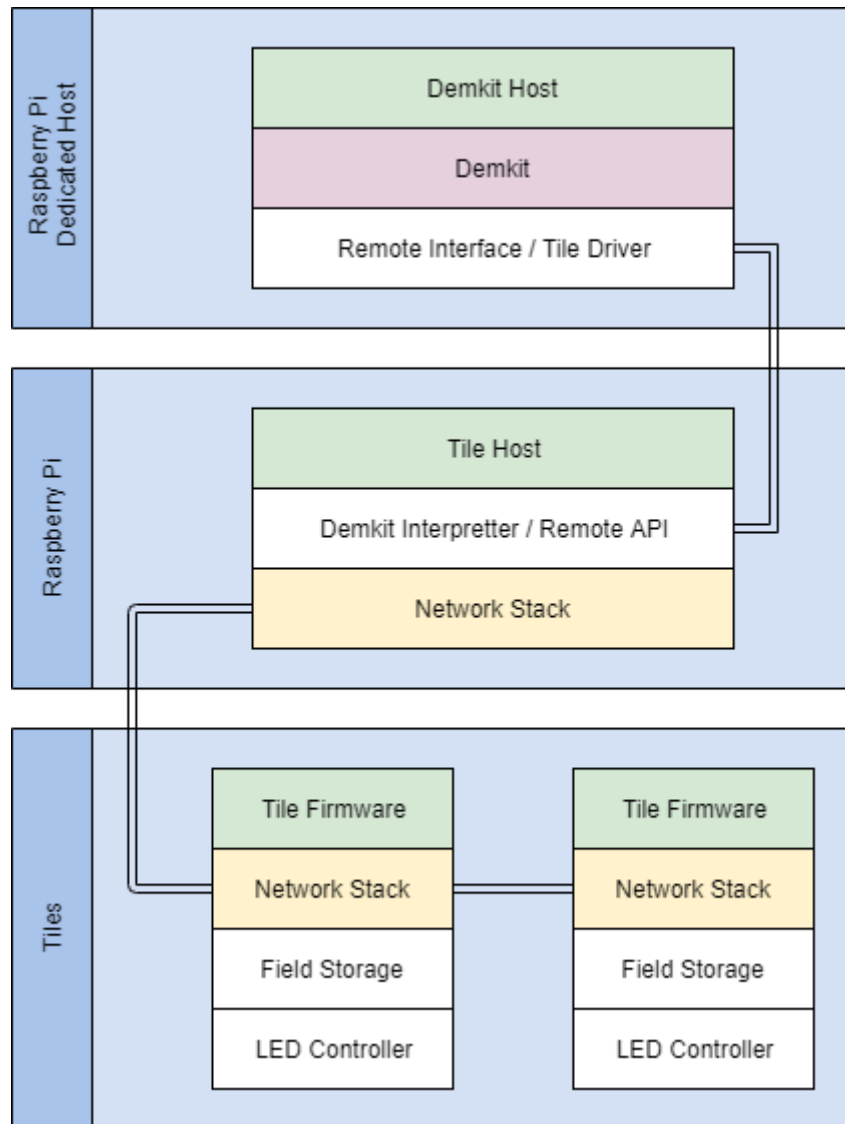


Figure 28: Architecture overview

The following sections will describe in more detail the different aspects of the software.

### 5.4.1 Software framework

The software framework used is called Arduino-CMake-NG ([Arduino-CMake-NG, n.d.](#)). It is a CMake based Arduino framework. This allows writing programs in any CMake compatible IDE (or no IDE at all) while not being limited to the Arduino IDE. The Arduino IDE is not suitable for larger projects and is not very flexible. By using CMake, organizing different parts of the software is easier and more mature IDEs like CLion can be used. These IDEs feature useful tools and code analysis which help reducing the coding time and potential bugs.



### 5.4.2 LED Control

To control the WS2812b, the FastLED library is used (FastLED, n.d.). This simplifies addressing the LEDs and gives convenient functions to work with color and animations of these colors.

### 5.4.3 Network

As the tiles can be connected in any shape, it's necessary to provide a mesh network between the tiles. A mesh network is a network where each node can communicate with others and together decide which route is best to send messages. (Rouse, 2019)

#### Physical Layer

As already described in 5.2, UART is chosen for data transmission. This section will describe in more detail why choices have been made. UART stands for Universal Asynchronous Receiver-Transmitter and is a simple way of transmitting serial data. UART is chosen because it's available both on the ATmega328P featured in the tiles and the Raspberry Pi featured as the host. UART also allows any kind of data transmission only limited by software.

Other systems for data communication include:

- I<sup>2</sup>C  
Not chosen as it's very susceptible for capacitance. (Hughes, 2018) It was expected that the combined capacitance of the tiles would be too high to perform reliable I<sup>2</sup>C communication at acceptable speeds. Especially since the whole networks connect each tile controller in parallel on one large bus. Using repeaters would not work since there can be loops in the connections which would keep the bus fixed to a state.
- SPI  
Not chosen as UART is essentially the same and considered easier to work with. In addition, SPI has a separate clock wire which means more connections.
- Wireless  
It was expected that the added complexity would make the solution infeasible for the duration of the project. In addition, the tiles would produce noise which could affect other systems.

As the communication is done over UART and because the used ATmega328P is limited in the amount of hardware UART drivers, a separate interrupt line (attention line) is added. This attention line allows combining all UART lines from the sides to 1 using multiplexers and demultiplexers. The attention line also makes it easier to signal the start and end of serial communication. This is important as limited serial communication can be received when FastLED is writing to the LEDs (Garcia, 2019).

The timings of the attention line are displayed below:

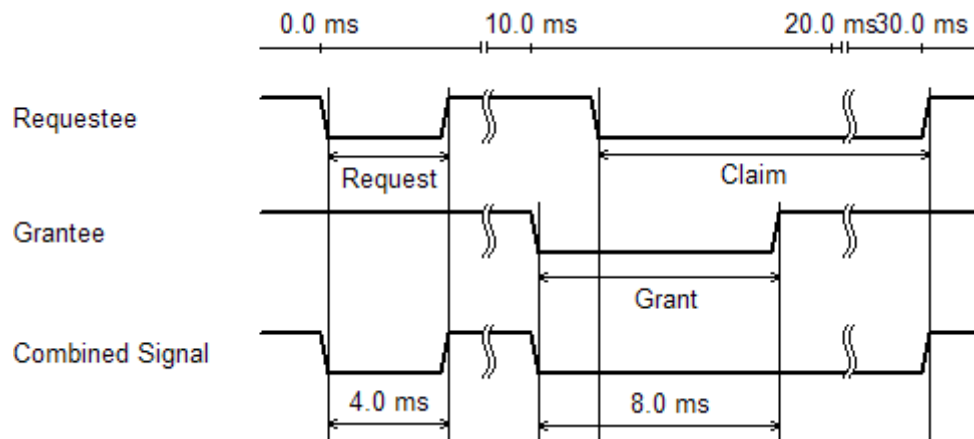


Figure 29: Timing diagram of the attention line

1. The requesting tile (Requestee) pulls the line to 0V for 4ms. This is long enough to let FastLED finish writing to the LEDs which allows the interrupt handler to catch the state of the line before it changes
2. After the 4ms, the attention line is released high and the requestee is waiting till the grantee tile is pulling the line to 0V.
3. At convenience of the grantee tile, it will pull the line to 0V for 8ms.
4. The requestee will check the line every 4ms for up to 10 times.
5. If the requestee senses the line is low, it will also pull the line low to indicate a claim.
6. Now the tile has attention till it releases the line and it can talk over UART.

## Datalink layer

On the UART, packets are sent which start and stop with magic bytes. These bytes signal the start, length and stop of the packet which provides basic detection of packet loss. Each byte only carries 7 bits to differentiate them from start and stop bytes.

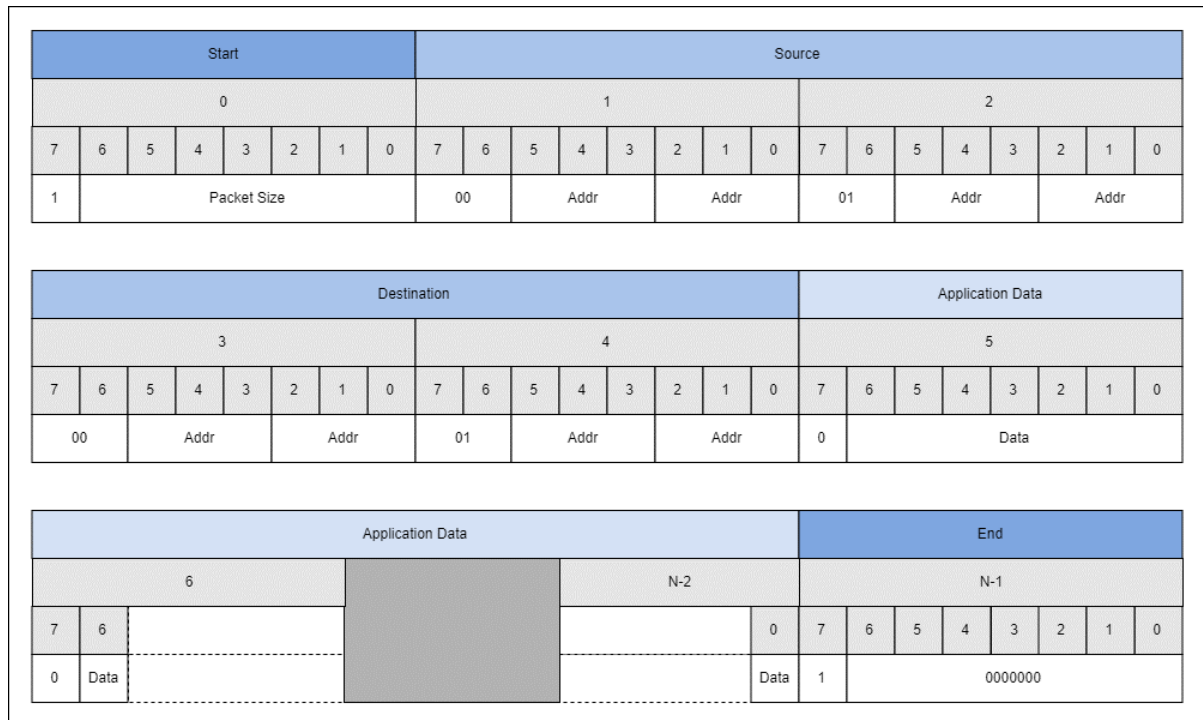


Figure 30: Packet construction

The packets contain the source and target address which are related to the sides of the tiles. This allows communication to any side. There are special packets that are used by the network which indicate tile connection and disconnection. Other packets indicate packet reception success or failure. And lastly there is a set of special configuration packets which allow setting parameters specific for the tile.

## Application Layer

The application layer specifies different messages for different usages:

Bits	Description
000 0000	Network status data
000 0001	Tile instructions

### Network messages

Network messages indicate changes to the network or signals for tiles related to network issues.

Bits	Description
000 0000	Tile disappeared
000 0001	Tile appeared
000 0010	Packet corrupt
000 0011	Packet received

### Tile instructions

Bits	Description
fff vvvv	Set field with code 'fff' to value 'vvvv'

The specific tile instruction packets consist of a field id and value. This allows a tile to configure several fields (for instance energy usage/production) to a specific value. Tiles can then animate LEDs based on the value that these packets contained. Multiple of these tile instructions can be appended which allows one packet to set multiple values to reduce overhead due to the start and stop frame.

## 5.4.4 Setup Demokit simulation

The host receives the combination (and topology) of the tiles that are connected and can create a model matching this setup. It's possible to dynamically change parameters of the model and composition (like different households or more solar panels).

### 5.4.5 Energy visualization

To visualize the energy consumption, two methods can be used:

- Modifying/intercepting Demkit devices to get the energy consumption and sending this to the tiles
- Reading already generated data from Influxdb

#### Modifying/intercepting

This method works by subclassing certain Demkit Devices and reading the consumption variable. This is a very intrusive but quick way to get some visualization.

Benefit	Drawbacks
Does not require Influxdb to run	Is highly dependent on how demkit works
	No fine-grained control on when and how much consumption data is received
	A simulation must run

#### Possible implementation

A dynamic python subclass which takes the actual subclass as argument is created. This subclass overrides the timeTick function to intercept the generated consumption data. By using a function which returns a class, it's possible to call it as "probeDevice(CurtDev, probeTarget, ...arguments...)". This data is then sent through an API to the tile host software (possibly TCP/HTTP/WebSocket) which translates it to the required field changes.

The Demkit software will ask the host to probe the tiles at the start to determine the topology and allow Demkit to create the model. Creation of the model can be done through the regular Demkit method or through the proof of concept "Demkit as a library". The latter method allows easier recreation of the Demkit simulation which would allow it to adapt to changes to the tile configuration while simulation is running.

#### Reading already generated data

This method works by asking Demkit which part of the simulation has been completed and next querying Influxdb for the interesting range of data.

Benefits	Drawbacks
No modification to Demkit necessary	Requires influxdb to run somewhere
Can attach to an already running demkit + influxdb	
Can play back pre-generated simulation data	

#### Possible implementation

A Demkit host making use of the proof of concept "Demkit as a library" provides a callback which performs a single simulation step and returns the time range of the simulated data. The tile host software first provides the attached tiles to Demkit through special callbacks for configuring the model topology (adding solar, wind, households, etc.). Next the tile host software calls the simulation callback to let Demkit populate Influxdb and get the simulated time range. The tile host software queries Influxdb to get the data and runs through the data at the correct speed, updating the tiles as the values change over time.

## 6. Realization

In this chapter the realization of the chosen concept will be shown. First the drawings will be discussed, this part consists of multiple iterations and test for the different components. After this the building process of the hardware will be summarized and the electrical components off the hardware will be explained.

### 6.1 Drawings of the components

Each tile consists of multiple sides, these sides contain the USB connectors, these USB connections are placed into inlays, so they can be easily swapped out when they break. Five of the six sides have the same block with USB inlay's, but one side has a special block in which there is place for the led diffuser which is used to show the energy flows. In Figure 32 and Figure 31, both the normal side as well as the side with space for the led diffuser is shown from multiple angels.

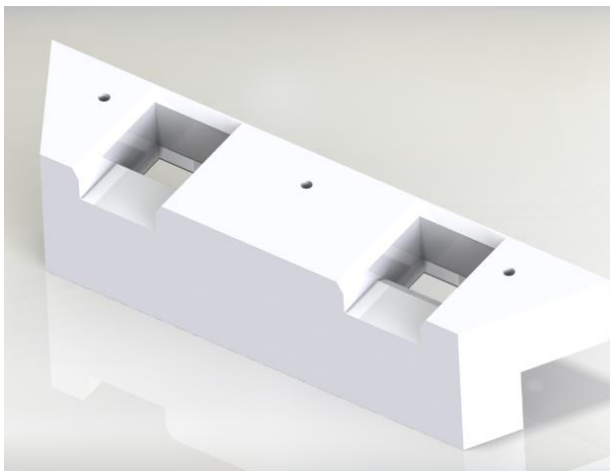


Figure 32: Side without space for diffuser.

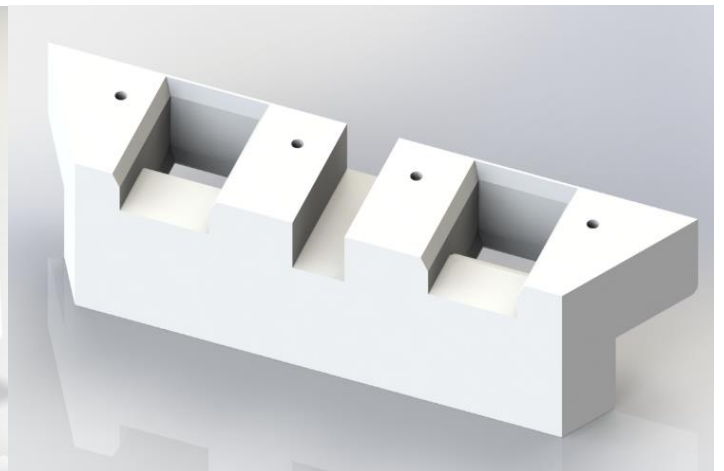


Figure 31: Side with space for diffuser.

In-between the tiles the connection blocks will be placed, in these blocks there will also USB connectors and a led strip that will show the grid. In the first designs the led strip was supposed to go on top of the connection block, this is shown in Figure 33, after this concept was produced an alteration was made to the design and the led strip was embedded into the connection block, this resulted into a better light effect and was chosen as final design which is shown in picture Figure 34 the bottom in which the USB's will be placed and where the cables will be managed is shown in picture Figure 35.

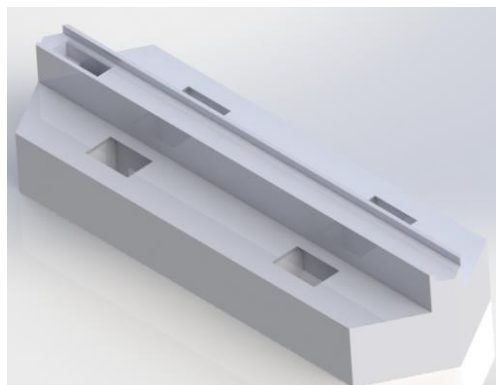


Figure 33: Connection block with led strip on top.

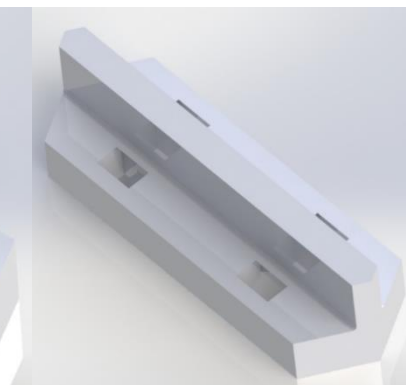


Figure 34: Connection block with led strip inside.

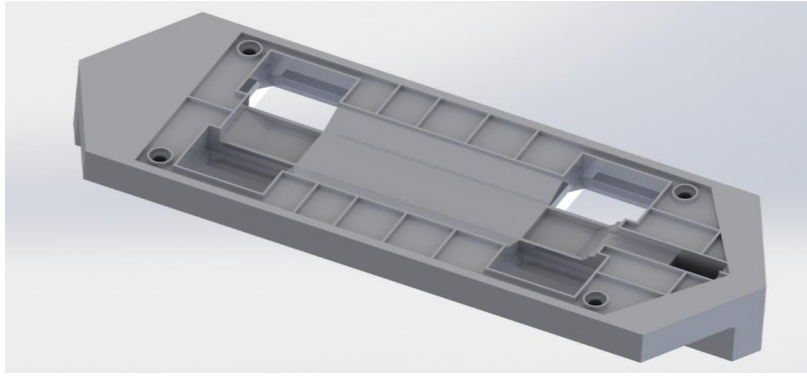


Figure 35: Bottom side of the connection block.

There are two different inlays for the USB's, one for the male USB's that are in the tiles and one for the female's that are in the connection blocks. Both USB's will be completely embedded into the inlays. On one side off the inlay's there is space to connect the cables. The inlay for the male USB is shown in picture Figure 36 and the female USB can be seen in picture Figure 37.

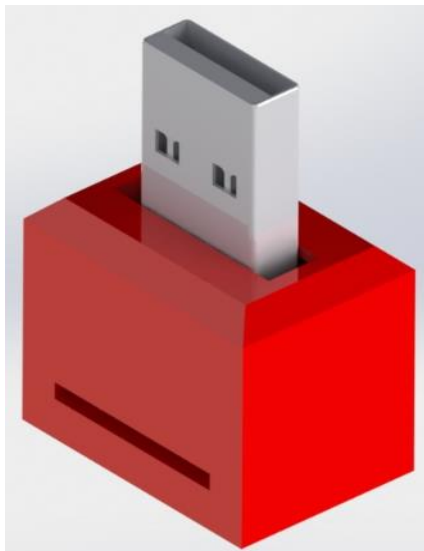


Figure 36: Inlay male USB.

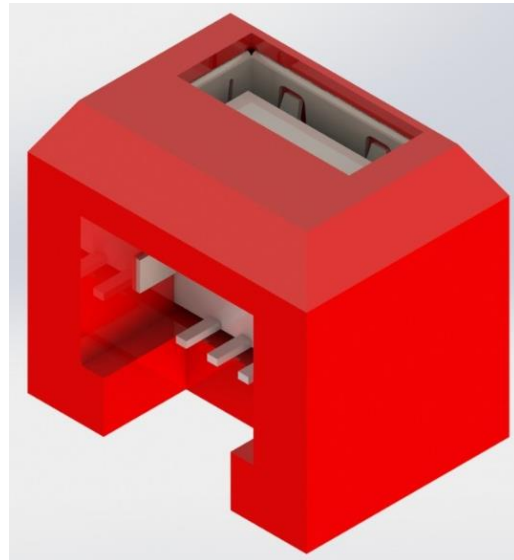


Figure 37: Inlay female USB.

To make the light effects that are on the tiles nicer led diffusers where designed to give a nice and uniform light strength. First the thickness off the diffuser had to be designed, this was done by making multiple concepts for the diffuser and testing them on how they diffused the light and what light strength could be accomplishment. In picture Figure 38 the different diffuser types that where tested can be seen. The one with the red circle around resulted in the best light effects and was chosen for the diffusers on the tiles and inside the connection blocks.

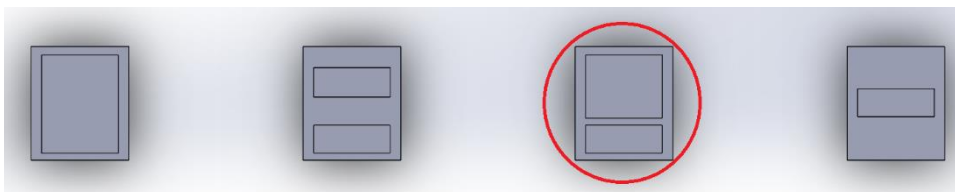


Figure 38: Different possibilities for led diffuser.

For all the parts 2D drawings can be found in appendix 11.4



## 6.2 Production of the tiles

On every tile there are 6 3D printed sides, 5 off these sides are like each other, in one side there is an extra space for the led diffuser. The production of each tile started with the 3D printing of these sides, together with the sides the inlays for the USB's and the led diffusers where printed.

The sides where printed with up to 8 pieces at the time on the printer as shown in Figure 39, for each tile this took about 20 hours for all the sides.

The inlays were also printed simultaneously, each inlay took about 30 minutes so for one tile this is 6 hours of printing. The bottom part of each inlay was printed first, when the printer reached a certain point it stopped so the USB's could be inserted into the prints. When all the USB's where inserted the printer continued with the top part off the inlay to ensure the USB's where completely embedded into the inlays. In the Figure 41 and Figure 40 the USB's can be seen while they are being printed.



Figure 39: Printing the sides of the tiles.

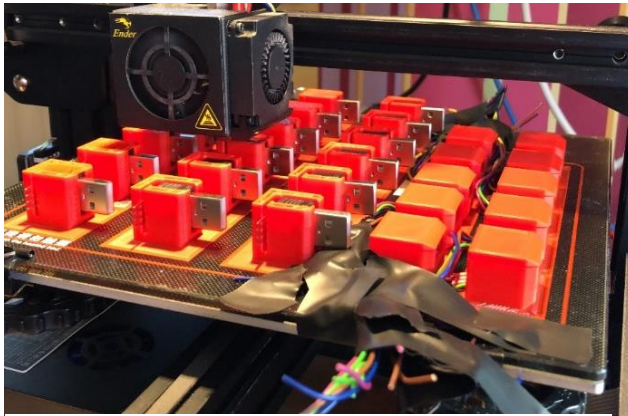


Figure 41: The USB's during printing.



Figure 40: The USB's during printing.

The led diffusers where printed in two parts, a top and a bottom part, in the bottom part the LEDs and cables are placed as seen in the bottom part off picture Figure 42. The top has a hollow space to ensure a nice diffusion of the light. Each led diffuser took about 6 hours to print for the top and bottom part combined.

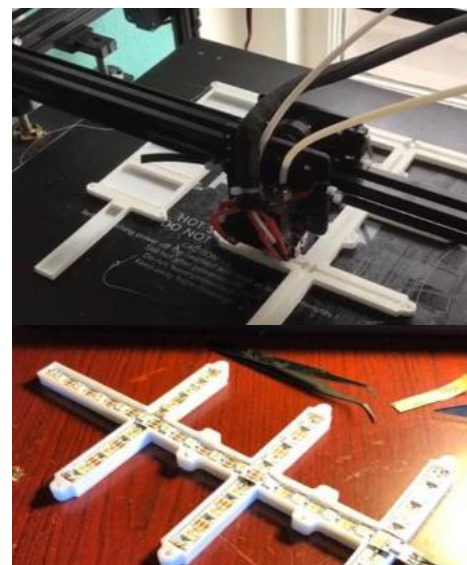


Figure 42: The led diffuser during printing (top) The led diffuser with led (bottom).



After all the 3D printer pieces were ready the top and bottom plate were made using the laser cutter, first the laser cutter engraved the pictures that were used to show the different components of the smart grid. After the engraving the machine cut out the hexagonal shape and the space needed for the led diffuser both the engraving and cutting can be seen in Figure 43.



Figure 43: Laser engraving and cutting off the tiles.

Now that all the components for the tiles are produced the assembly process can begin, first all the sides are connected to the bottom plate, after that the male USB connectors are placed into the sides, this can be seen in Figure 44 on the left. When all sides are filled with USB's the top plate with the led diffuser is screwed to the sides, this is done with 3 screws per side as seen in Figure 44 in the middle and on the right. Now the top plate is fixed the bottom plate can be taken off to get to the electronics, the assembly of the electronics will be discussed further on in this chapter.

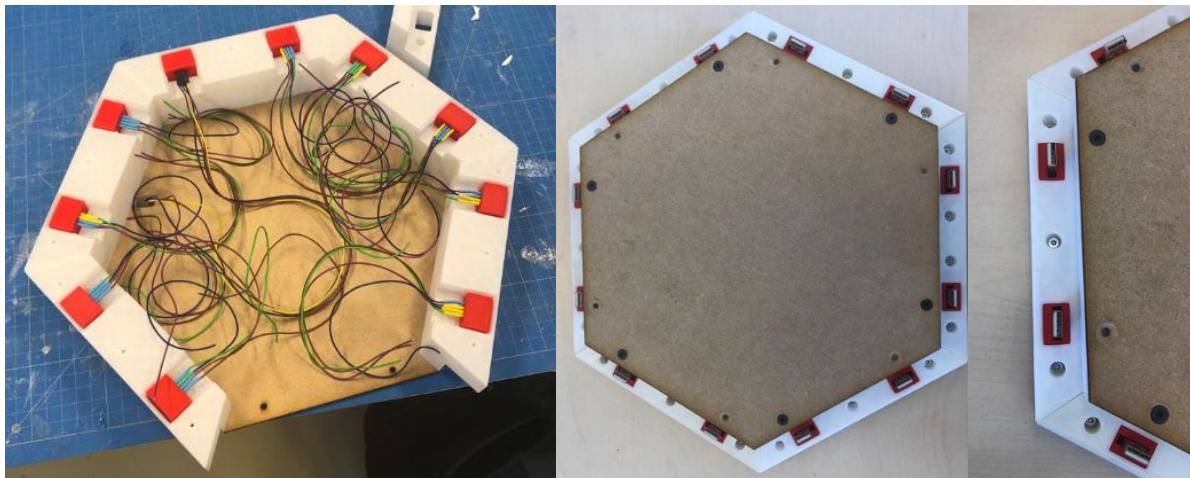


Figure 44: The assembly process of a tile.

The connection blocks were also made using 3D printing, during the printing the printer stops at a certain height, so the led strip can be inserted into the print. The printing of one connection block can be seen in Figure 45, each block took about 6 hours to print.

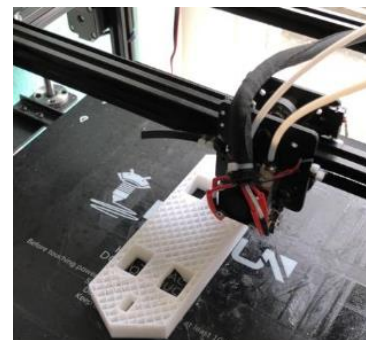


Figure 45: Printing a connection block.

After the printing of the connection block was completed the female USB inlays were inserted, the cables of the female USB's were crossed over and connected with the led strip that is in the top of the connection block, this can be seen in Figure 46, the USB's with the blue circle around them are connected together. The crossover is needed to connect the right cables inside the tiles with each other.

To close of the connection blocks a bottom plate was made using the laser cutter. This can be seen in Figure 47 on the left. On the right of Figure 47 the connection block can be seen from the top with the USB's inserted.

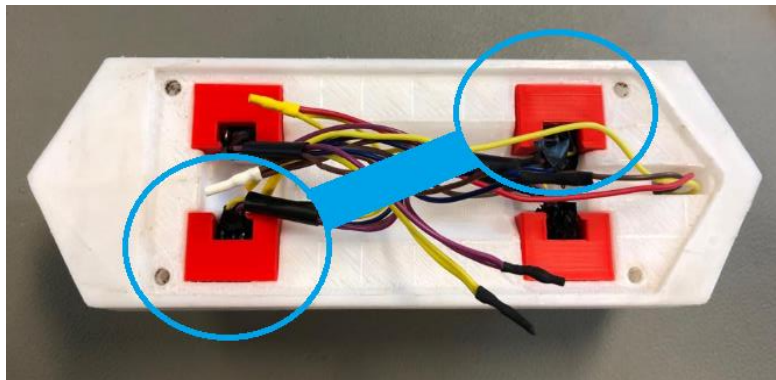


Figure 46: Connection block and cable layout.

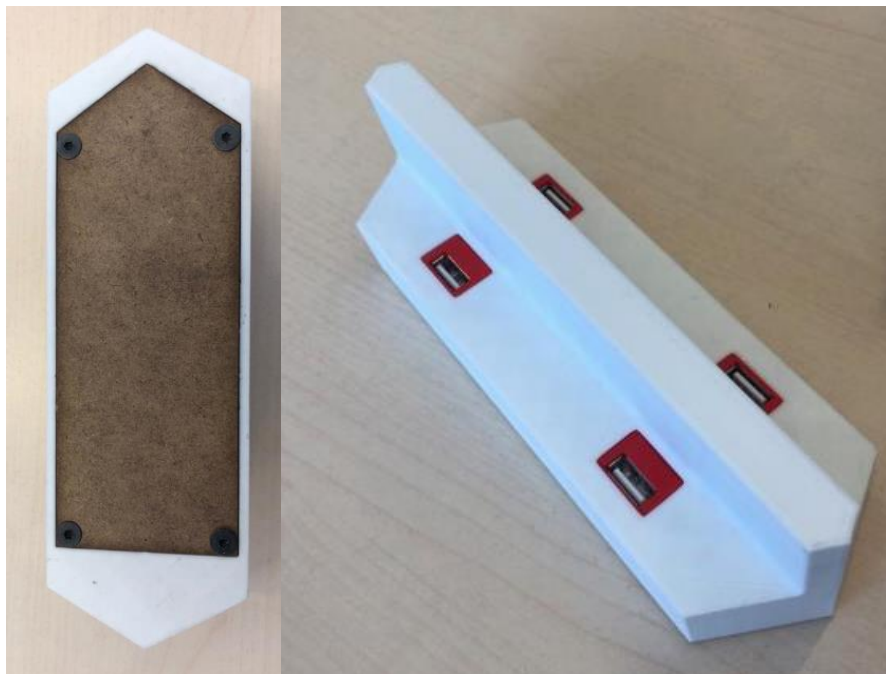


Figure 47: A connection block from the bottom (left), and from the top (right).

## 6.3 Electronics Realization

The realization started with ordering the components and the PCB which was ordered from AllPCB. The figure below shows all steps the PCB underwent during the manufacturing of it at AllPCB.

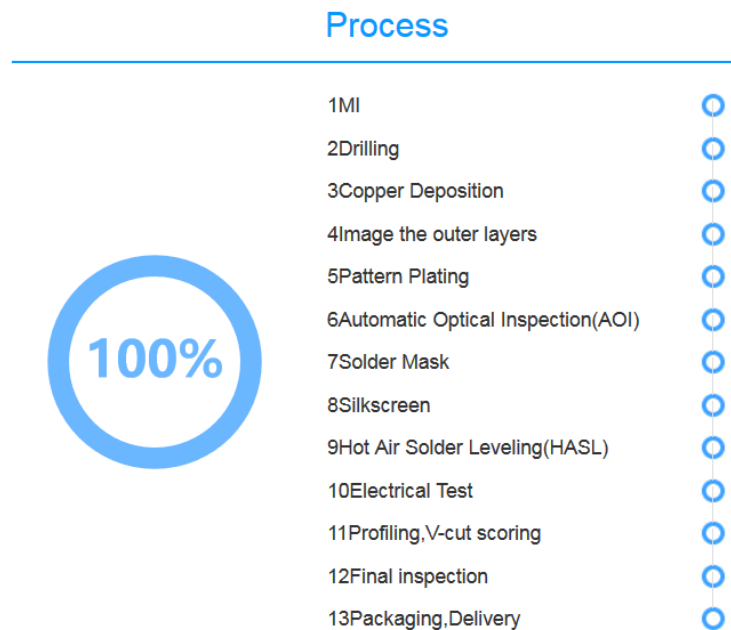


Figure 48: Process of creating a PCB

After receiving the PCB, the first step was a visual inspection of all boards. The top and bottom of the bare PCB can be seen in the figure below:

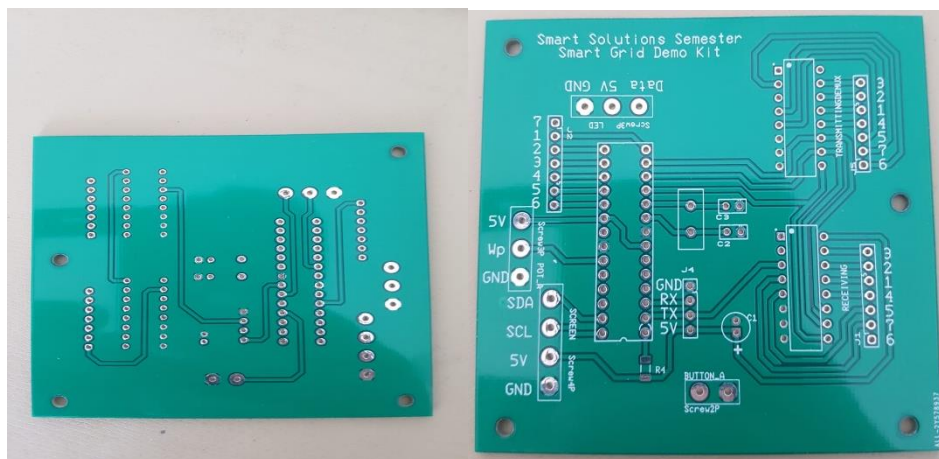


Figure 49: The produced PCBs

During the inspection it was checked if all holes and pads were properly placed and connected on the PCB. After this all components were fitted onto the board to check if the footprints matched. The last step of the assembly of the PCBs was soldering them.

The soldering was followed by placing the chips onto the sockets and measuring all connections with the multimeter. The assembled PCB can be seen in the figure below.

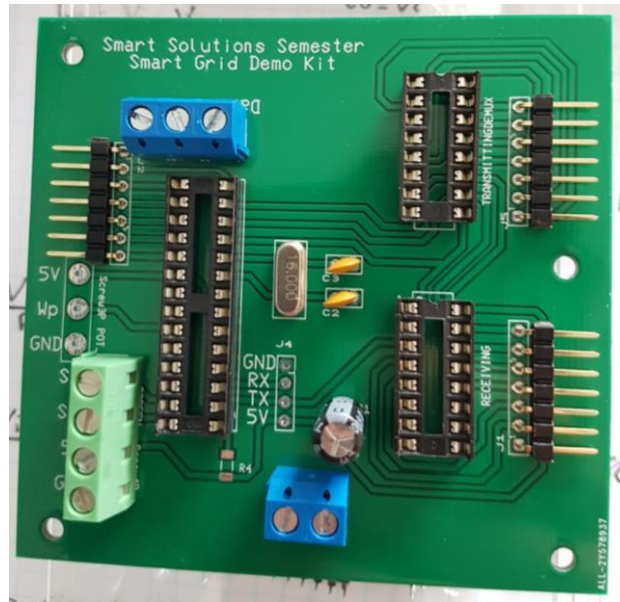


Figure 50: Finished board

All wires were soldered to the USB connector as can be seen in the figure below. After this the USB connectors are placed in the tiles and the assembly of all electronics in the tiles started.

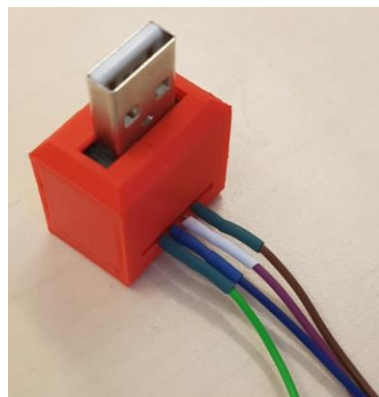


Figure 51: Male USB connector block

The wires from the USBs are connected according to Figure 12 in the low-level design. All +5V, ground and data grid lines are connected to each other. The RX, TX and interrupts lines are connected to the PCB using female headers. Using this, all wires can be disconnected from the PCB. The PCB is connected to the +5V and GND and to the digital LED strip of the tile itself. The grid block USB wires are soldered cross links as mentioned earlier. Heat shrink tubing is used for every connection that is soldered, so it will stay strong for a long time.

Now all the connections are soldered and connected to the PCB. The PCB can be installed into the tile using the screw connectors.

There are two limitations for the expandability due to the thickness of the cables used and the used buck converter. The thickness of the cables gives a maximum amount of current they can carry and when too much current flows through them, the wires can melt. In the design this problem is minimized by using 2 power lines instead of one. For the five tiles which are used now, this is not a problem. But when using a lot more tiles, this could be a problem. The used buck converter can only generate a maximal current of 2 amperes. When using more tiles, more current will be drawn from the converter. So, for more tiles, this converter must be replaced by a better one.



## 6.4 Software

In this section the realization of the software is discussed. Software is being tracked in a version control system called Git.

### 6.4.1 Software framework

The suggested software framework is used but since it lacks configuration options of the programmer choice, a custom command to upload to the boards is used.

The reason for choosing a specific programmer is to bypass the regular Arduino bootloader. One of the interrupt pins is Arduino pin 13, which gets toggled by the bootloader on startup. This interferes with the communication protocol. When another programmer is used, the bootloader can be overwritten with the tile software which prevents this toggling of pin 13.

### 6.4.2 Physical Layer

The physical layer attention line part is implemented as described in the manual and verified with running on the hardware.

### 6.4.3 Datalink and Application Layer

A more extensive documentation of the data and application link has been written which allows direct implementation of the protocol in code.

### 6.4.4 Demkit Simulation

A proof of concept Demkit model has been created. Alterations to Demkit were made to support using Demkit as a library. This is done as the regular way of working with Demkit is reverse in thinking:

It's much more common for a system like Demkit to work as a library, giving building blocks for models. However, Demkit works different. It imports the model into itself and runs it. This might have some benefits but also brings serious drawbacks. One of the drawbacks is that it's not possible for IDEs to resolve all symbols. This means that a lot of syntax errors are shown which are not true.

Therefore, an attempt was made to turn Demkit into a library. This involved a simple step; all python modules should import themselves relatively. This allows Demkit to become relocatable, allowing it to be installed wherever users want it. Next Demkit was placed inside the model directory, which made all Demkit building blocks accessible through `'from demkit.<module> import <class>'` which also allowed the IDE to understand the code.

Next the example model was transformed into this new way which made it possible to dynamically add/remove model parts (e.g. complete removal of solar panels or storage). It also made it easier to reference settings from models (changing solar panel type during simulation). Which ultimately allows changing the model to reflect the tile state.

## 6.4.5 Energy visualization

### The current software

The software that is currently used on the tiles is a simple demo program. For each tile a different program was written together with a program to control the center grid.

The programs on the tiles control the LEDs that are on that specific tile and in the center tile there is an extra PCB which controls the LEDs in the connection blocks that represent the grid. The programs for the LED control were written in the Arduino IDE and make use of a standard Arduino library called FastLED. With this library it was possible to address each led inside a led strip individually. Below there are 2 lines from a program to control the LEDs.

```
leds[1] = CRGB(0,128,0);  
FastLED.show();
```

With leds[1] the specific led is targeted, with CRGB(0,128,0) the specific color and brightness of this led is specified. After the specification of the led and color the line FastLED.show turns the led on. In appendix [11.5] one whole program can be found.

### Future software

An additional proof of concept of the Demkit modification is made in co-operation with Amin Amin Ahmed Mohamed Abdalla Elomerabi.

It defines a special Probe class generating function which allows any LoadDevice subclass to be subclassed. This allows the class to provide all the features necessary by the Device classes but intercept the timeEvent which gives access to consumption data. A special driver class is also provided which can send this data to the tile hardware and can be connected to the Probe classes generated from the previous function.

This allows a simulation to update, in real-time, the visualization as the simulation runs. The downside of this method that it's running at the simulation speed which can vary depending on the difficulty of the model and complexity of the predictions in Demkit

## 7. Results and testing

---

### 7.1 Electronics testing

The testing of the electronics started with a visual inspection of the PCB. There was looked if every connection to the PCB look properly. The traces of all the connections were already tested with the mustimeter during the realization. To check if there are any short circuits in the board a +5V supply is connected. All the bards passed this test, meaning there are no short circuits in the PCB.

The next test was checking the functionality of the digital led strip connector. This was done by connecting a LED strip to the board and running a test program for the LEDs. The result can be found in the picture below:

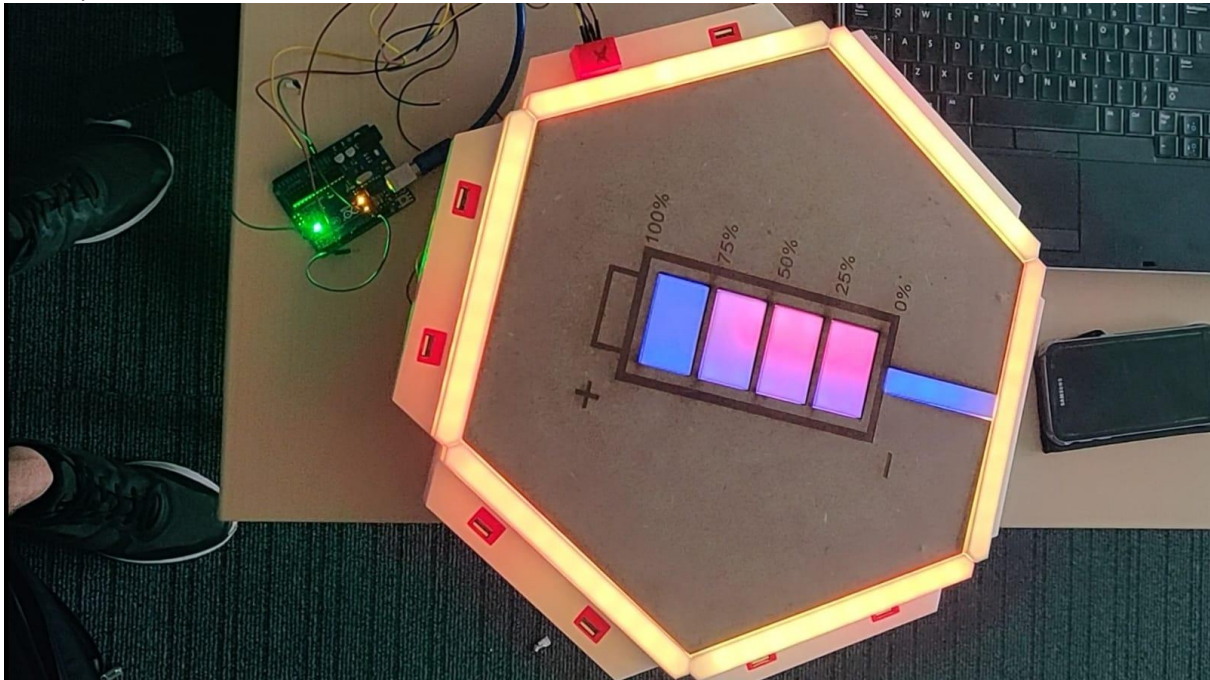


Figure 52: Running LED test program

The LED strip signal worked as expected, meaning the LED strip communication works properly. The last test for the PCBs was checking the functionality of the communication part. For this, interrupts were generated to simulate an incoming signal from another tile. The system had to switch both multiplexers to the same and corresponding channel of the interrupt. This was measured for each board and for each interrupt using an oscilloscope. An oscilloscope result can be seen in the figure below. It shows the interrupt which is generated by pulling the interrupt line low and how the multiplexer switches to the corresponding channel of the interrupt.

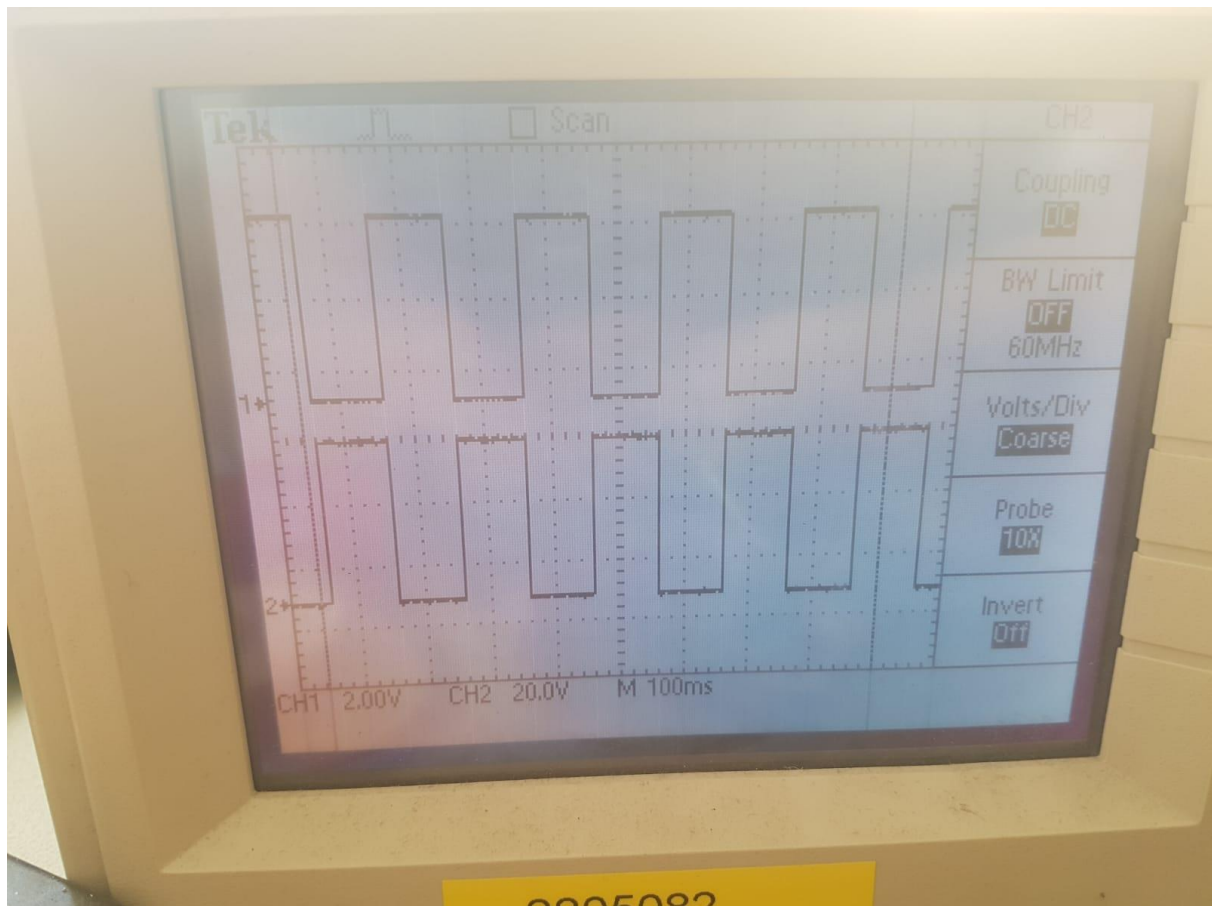


Figure 53: Response to interrupt line



The table below shows that all boards passed this test. The result from the test was that after each interrupt the corresponding channel was linked by both the RX and TX multiplexer.

INT_LOW	MUX_RX channel	MUX_TX channel	PCB successful
<b>1</b>	1 linked	1 linked	1,2,3,4,5
<b>2</b>	2 linked	2 linked	1,2,3,4,5
<b>3</b>	3 linked	3 linked	1,2,3,4,5
<b>4</b>	4 linked	4 linked	1,2,3,4,5
<b>5</b>	5 linked	5 linked	1,2,3,4,5
<b>6</b>	6 linked	6 linked	1,2,3,4,5
<b>7</b>	7 linked	7 linked	1,2,3,4,5

## 7.2 Software

### 7.2.1 Result

A large part of the software has been written and for parts that could not be finished, a proof of concept exists instead. All features that are implemented work as designed.

As indicated earlier, the FastLED driver needs special care as interrupts are disabled during controlling the LEDs. This also includes receiving serial data. Hence why it's important to not control LEDs while doing any serial communication to reduce the risk of packet losses.

### 7.2.2 Testing

Tests have been performed to verify the software can handle the attention line.

The software by default tunes into the 7<sup>th</sup> side which is used by the raspberry pi. As the software for each tile starts with an attention request on each side without any data transmission, it's possible to verify correct handling of the attention protocol with the following process:

1. Connect two tiles through their attention pins
2. Probe the attention line and the (de)multiplexer selection lines of the receiver
3. Note that the (de)multiplexer is selected by '111'
4. Reset the sending tile
5. The tile now probes all sides and the request protocol should eventually be visible on the scope
6. In the "claim" time of the attention protocol, observe the correct selection is provided to the (de)multiplexer according to table (insert reference to electronics)
7. After the "claim" time of the attention protocol, the selection lines should revert to '111'

## 8. Conclusion

---

This project consisted of doing research into and designing a smart grid demo kit. For this it is important to have a product which is clear for users who have little knowledge about smart grids. After doing a lot of research and comparing different concepts, the build of the chosen concept started. The concept consisted of hexagonal tiles, which were made using laser cutting and 3D printing. The tiles can be connected to each other using special grid blocks which form the central power grid. Both the tiles as the grid blocks have USB connectors which function as an electrical and mechanical connection. The use of USB connectors also makes the system easy to use for the users.

The five tiles (houses, wind power, solar power, factories and energy storage) all have LED strips on top of them to show the direction of the energy flowing. The grid blocks light up to show that they are at the central power grid. In the final product it is not yet possible to let the tiles communicate with each other. The hardware is ready to run be communication, but the software is not finished yet. For now, the tiles run standard programs to show the flow direction of the energy, but the tiles cannot react to changes elsewhere in the system. This gives users the idea of how smart grids work, because the flows of the energy are visible.

The tiles itself are decent and solid, because they are made from wood and plastic. The USB connectors are also very durable in this design. The electronics consist of all connections between the USB connectors and a PCB for each tile. This PCB makes the communication between the tiles possible. The kit also features a box where all tiles can be placed into for transportation. This makes it easy to take it to an information market.

## 9. Recommendations

---

The recommendations will be divided over the 4 different design categories; General, Mechanical, Electrical and software. For each categories some advices will be given and the steps that are possible in a further project will be highlighted.

### 9.1 General recommendations

In the general recommendations the ordering, building and assembling process will be discussed as well as some recommendations for a future group composition.

For the ordering process it is recommended to look at other suppliers for the components, ordering components in China could reduce the costs per tile dramatically, about €25 per tile instead of €100 which is the current price.

### 9.2 Mechanical recommendations

A part of the mechanical engineering is the transport box for the demokit, this box has some room for improving.

- 1.) A soft material to protect the tiles better during transport.
- 2.) Wheels to make it easier to transport for over larger distances.

Also the connection block can be improved slightly, the problem with this component is that the LED strip is embedded in the 3D printed block this makes it impossible to replace the LED strip if it breaks, this makes the connection block useless when the LED strip breaks. If the LED strip wasn't embedded into the print itself this would make it possible to replace.

### 9.3 Electrical recommendations

The electrical part of the project has some scope for improvement. The electrical recommendations are listed below:

- Smaller PCB

The PCB used in the project was only the first prototype and because of this there is still room for improvement. The microcontroller, traces and clearances could be made smaller to decrease the total size of the PCBs significantly.

- Better power supply

The used buck converter as power supply got quite hot during operation of the system. This limited the number of connected tiles. This was not a problem for this project, because it only used five tiles. But when expanding the system, it is important to upgrade to a better power supply.

- Improving uploading header on PCB

A programming header was placed on the PCB, but it missed one pin connection to be able to upload code to the microcontroller. In a future PCB design, this pin can be added to the upload header, making it easier to program the PCB.

- Better cable coloring

During the soldering of the wires to the USB connectors some wire colors were not available. It would have been better to check for enough wire of each color before the start of the soldering process.

## 9.4 Software Recommendations

For continuation of this project, there is a list of recommendations that can be considered. Some relate to almost finished work and some relate to ideas to extend it beyond what was originally intended.

- Finish the network protocol so tiles can communicate to each other  
Currently only a theoretical description of the protocol exists which can be
- Writing interface software to control the tiles from the center tile
- Implementing the suggested bridge between Demkit and the tile controlling software
- Write a realistic, customizable model for Demkit to accurately simulate the represented smart grid
- Create an interface to control parameters of the tiles. Recommended is to make a web-based frontend. This would make it compatible with computers, laptops, tablets and phones without much additional work
- Apply gamification to the simulation:
  - Possibly score the user based on:
    - Grid Stability
      - Grids should generate 0 net energy
    - Grid Cost
      - Grids cannot be unlimited in price. E.g. larger battery storage could make a grid more stable but costs much more
    - Household happiness
      - Turning off certain appliances can help power consumption, but turning off a TV will not have good user experience
  - Suggest the user improvements or give explanations about the score change
    - If a battery is removed, solar power can only be used during daytime

## 10. Literature

---

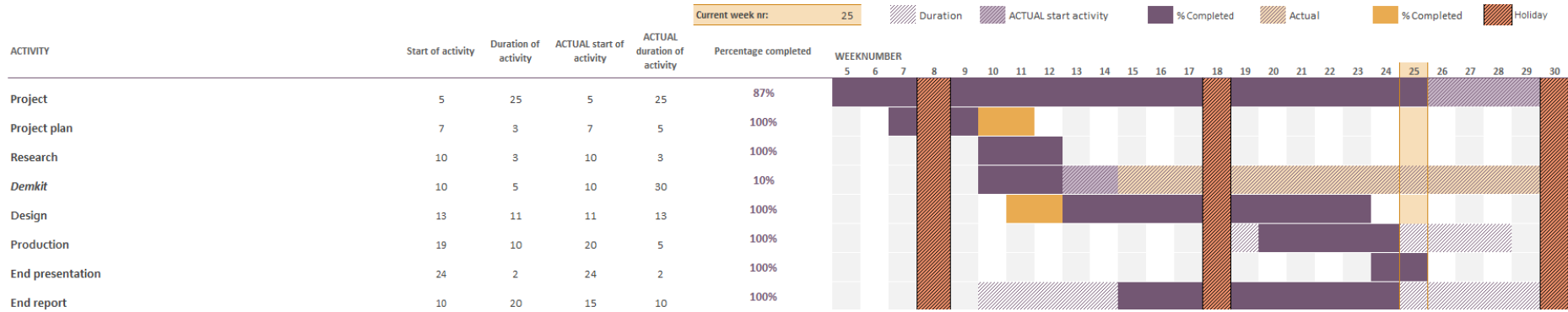
- Arduino-CMake-NG*. (sd). Opgehaald van <https://github.com/arduino-cmake/Arduino-CMake-NG>
- FastLED*. (sd). Opgehaald van <https://github.com/FastLED/FastLED>
- Garcia, D. (2019, January 20). *Interrupt Problems*. Opgehaald van <https://github.com/FastLED/FastLED/wiki/Interrupt-problems>
- Hughes, M. (2018, June 22). *I2C Design Mathematics: Capacitance and Resistance*. Opgehaald van All About Circuits: <https://www.allaboutcircuits.com/technical-articles/i2c-design-mathematics-capacitance-and-resistance/>
- IEEE ISPLC . (2007, 3 26). *TELEGESTORE PROJECT*. Opgehaald van ISPLC: <http://isplc2007.ieee-isplc.org/docs/keynotes/rogai.pdf>
- International Energy Agency. (2013, 2 19). *Technology Roadmap Smart Grids*. Opgehaald van Slideshare: <https://www.slideshare.net/abekas/iea-technology-roadmap-smart-grids>
- Munuera, L. (2019, 6 5). *Smart grids*. Opgehaald van IEA: <https://www.iea.org/tcep/energyintegration/smartgrids/>
- Rouse, M. (2019, April). *Mesh network topology (mesh network)*. Opgehaald van IoT Agenda: <https://internetofthingsagenda.techtarget.com/definition/mesh-network-topology-mesh-network>
- Smart grids*. (2019, 4 1). Opgehaald van smartgridsinfo: <https://www.smartgridsinfo.es/smart-grid>
- Smith, S. (2018). *SMART GRIDS TO SAVE CITY DWELLERS \$14BN IN ENERGY COSTS BY 2022*. Opgehaald van juniperresearch: [https://www.juniperresearch.com/press/press-releases/smart-grids-to-save-city-dwellers-\\$14bn-in-energy](https://www.juniperresearch.com/press/press-releases/smart-grids-to-save-city-dwellers-$14bn-in-energy)

11. Appendix

[Here you can find the updated planning](#)

11.1 Planning and scheduling

Projectplanner



## 11.2 List of requirements

### 11.2.1 Functional requirements

- Interaction with the user through Gamification (Visual and by App)
  - Controllable from a mobile device
  - Controllable standalone
- Each tile is a representation of a possible part of the smart grid
- Time simulation for a whole day (24 hours)
- At least four tiles including:
  - Must have
    - Houses
    - Solar/Wind park
    - Industry
    - Storage
  - Could have
    - Bio energy factory
    - Offices
    - City center
    - Second type of houses
- Fast connections between the tiles (within 10 seconds)
- Durable
  - Connections/tiles shouldn't break during normal usage
  - System should withstand transport in designed packaging
- Dummy proof system
- Visual presentation of energy flow
- Understandable for normal developed children from ten years and older
- Sustainable materials to make the transporting package
- Tiles should be identifiable, and the client has to approve the design
- Should use Demokit as functional base

### 11.2.2 Gamification

- Give users tasks to optimize the smart grid (rearranging tiles, choosing different tiles)
  - App gives scenarios (predefined different external environment and tile behaviors) for users to solve
  - App shows and tracks global grid performance
    - Improvement in grid performance gives positive feedback



- Reduction of grid performance gives hints and causes
    - Performance of a given scenario is given a score which can be compared to others
- Connecting tiles gives sensory feedback (audio/visual and/or touch)

### 11.2.3 Technical requirements

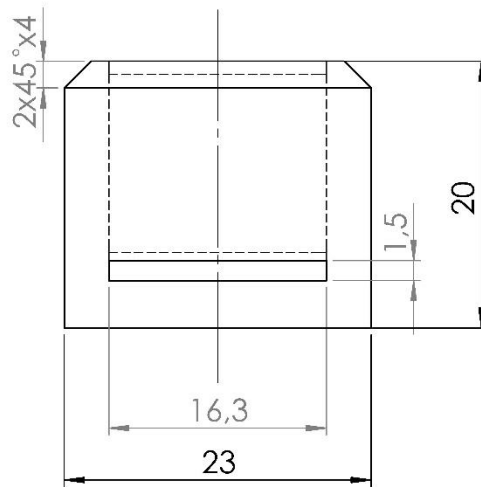
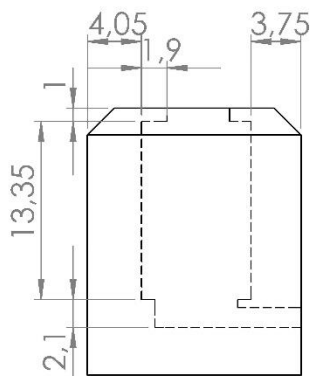
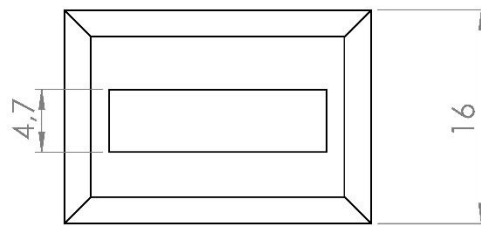
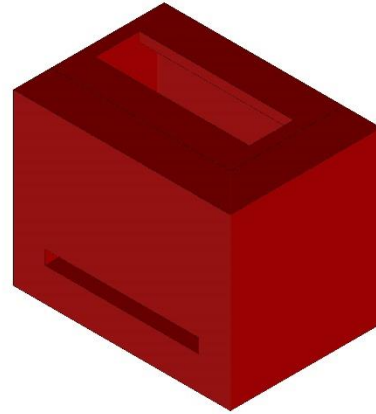
- Tile sizes about 200 \* 200 mm and maximum 400 \* 400 mm
- Usage of safe voltage (below 15 Volts) and isolating material
- RGB LED's to show the energy flow
- Electronic connections between the tiles
- Laser cutter, 3D printer or CNC to make the tiles
- Standalone unit (no WiFi needed)
- Switches that activate certain functions (decided further)
- Integrated screen to monitor efficiency and score of the App
- Use snap on connections to connect the tiles
- Show energy flow of:
  - House power grid (and in a street)
  - Solar panels
  - Wind turbines
  - Power plants
  - Energy storag

### 11.3 Scoring criteria

Score criteria	Sub criteria	Weight factor [1-5]	Sub criteria weight factor [1-3]/[1-2]	Concept 1	Concept 2	Concept 3	Concept 4
<b>Logical shape</b>		<b>5</b>		83	88	72	87
	Organized/Logical		1	6	5	3	4
	Self-explaining		3	8	8	5	9
	Relation to reality		2	4	5	7	6
<b>Expandability</b>		<b>3</b>		42	43	54	61
	Future use		2	3	5	6	7
	Different configuration		2	7	6	7	8
	Versatile		2	5	5	6	6
<b>Possible energy flow representation</b>		<b>4</b>		64	77	59	64
	Relation to reality		2	5	7	9	7
	Organized/Clear		2	8	8	4	6
	Possibility to show more than 1 energy flow		1	3	6	3	3
<b>Producibility</b>		<b>1</b>		9	10	11	12
	Reusable shapes		2	5	5	6	7
	Shape complexity		1	3	4	3	4
<b>Transportability</b>		<b>2</b>		33	35	42	50
	(Uniformity in) size		1	3	3	4	6
	Possibility to take it with you		3	6	7	8	10
	Weight		2	5	5	5	6
Scoring is from 1-4, 1 is insufficient, 2 is sufficient, 3 is good, 4 is excellent							

## 11.4 2D-Drawings

### 11.4.1 USB Inlay Male



Am. Projection

DIMENSIONS  
IN MILLIMETERS

DEBUR AND BREAK SHARP EDGES  
TOLERANCES ACCORDING TO: ISO 2768-mK

Finish:

Material:

Description:

	Name	Signature	Date
Drawn:	Vincent Scholten		19-6-2019
Checked:	Justin Markvoort		19-6-2019
Class:	Smart Grid Demokit	StudentNo.	438528



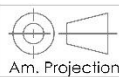
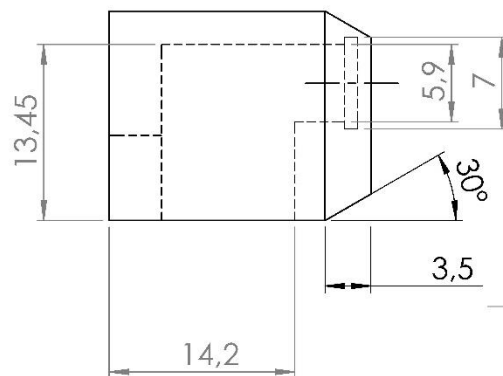
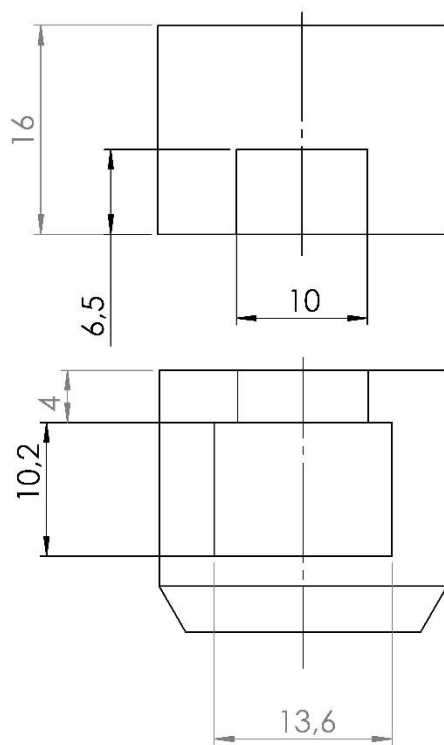
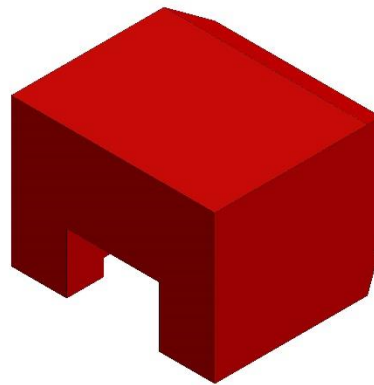
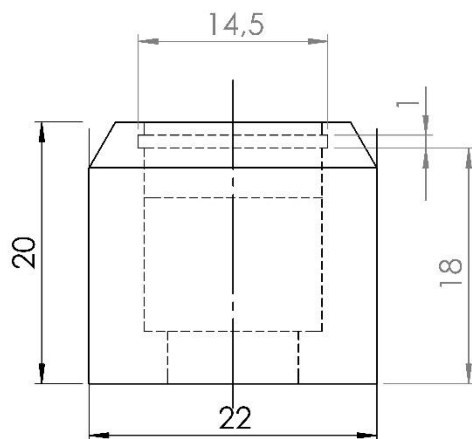
Address: M.H. Tromplaan 28, 7513 AB Enschede, The Netherlands  
Postal address: P.O. Box 75000, 7500 KB Enschede, The Netherlands  
Phone: +31 53 4871111 Fax: +31 53 431 2233 Web: www.saxion.nl

Drawing number:	Revision:	Format
<b>Inlay male USB</b>		<b>A4</b>

Weight:	SCALE: 2:1	SHEET: 1 OF 1
---------	---------------	------------------

THIS DOCUMENT AND ALL INFORMATION AND DATA DISCLOSED HEREIN OR HERewith ARE THE CONFIDENTIAL AND PROPRIETARY PROPERTY OF SAXION AND ARE NOT TO BE USED, REPRODUCED OR DISCLOSED IN WHOLE OR IN PART BY OR TO ANYONE WITHOUT THE WRITTEN PERMISSION OF SAXION.

### 11.4.2 USB Inlay Female



Am. Projection

DIMENSIONS  
IN MILLIMETERS

DEBUR AND BREAK SHARP EDGES  
TOLERANCES ACCORDING TO: ISO 2768-mK

Finish:

Material:

Description:

	Name	Signature	Date
Drawn:	Vincent Scholten		19-6-2019
Checked:	Justin Markvoort		19-6-2019
Class:	Smart Grid Demokit	StudentNo.	438528



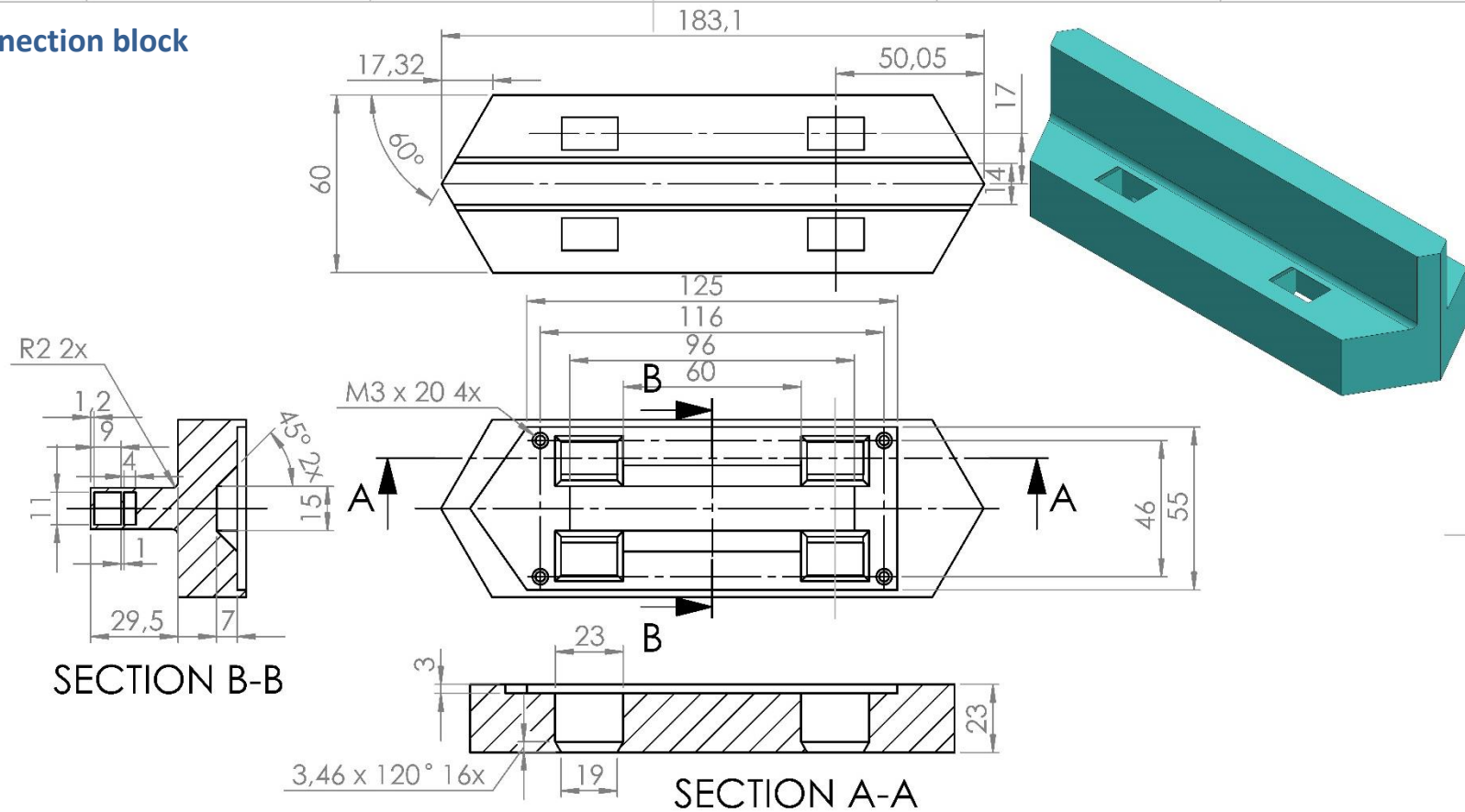
Address: M.H. Tromplaan 28, 7513 AB Enschede, The Netherlands  
Postal address: P.O. Box 75000, 7500 KB Enschede, The Netherlands  
Phone: +31 53 4871111 Fax: +31 53 431 2233 Web: www.saxion.nl


Drawing number:	Revision:	Format
<b>Inlay female USB</b>		<b>A4</b>

Weight:	SCALE: 2:1	SHEET: 1 OF 1
---------	---------------	------------------

THIS DOCUMENT AND ALL INFORMATION AND DATA DISCLOSED HEREIN OR HEREIN ARE THE CONFIDENTIAL AND PROPRIETARY PROPERTY OF SAXION AND ARE NOT TO BE USED, REPRODUCED OR DISCLOSED IN WHOLE OR IN PART BY OR TO ANYONE WITHOUT THE WRITTEN PERMISSION OF SAXION.

### 11.4.3 Connection block



 Am. Projection		DIMENSIONS IN MILLIMETERS		DEBUR AND BREAK SHARP EDGES TOLERANCES ACCORDING TO: ISO 2768-mK		Finish:			
				 Address: M.H. Tromplaan 28, 7513 AB Enschede, The Netherlands Postal address: P.O. Box 75000, 7500 KB Enschede, The Netherlands Phone: +31 53 4871111 Fax: +31 53 431 2233 Web: www.saxion.nl		Material:			
						Description:			
Drawn:	Vincent Scholten	Signature	19-6-2019			Drawing number:	Revision:	Format	
Checked:	Justin Markvoort	Signature	20-6-2019			<b>Connection block</b>		<b>A4</b>	
Class:	Smart Grid Demokit	StudentNo.	438528			Weight:		SCALE: 1:2	SHEET: 1 OF 1
THIS DOCUMENT AND ALL INFORMATION AND DATA DISCLOSED HEREIN OR THEREWITH ARE THE CONFIDENTIAL AND PROPRIETARY PROPERTY OF SAXION AND ARE NOT TO BE USED, REPRODUCED OR DISCLOSED IN WHOLE OR IN PART BY OR TO ANYONE WITHOUT THE WRITTEN PERMISSION OF SAXION.									

## 11.5

```
#include <FastLED.h>
```

```
//Programm for the windmills  
// How many leds in your strip?
```

```
#define NUM_LEDS 33
```

```
// For led chips like Neopixels, which have a data line,  
ground, and power, you just  
// need to define DATA_PIN. For led chipsets that are SPI  
based (four wires – data, clock,  
// ground, and power), like the LPD8806 define both  
DATA_PIN and CLOCK_PIN
```

```
#define DATA_PIN 4
```

```
#define CLOCK_PIN 13
```

```
// Define the array of leds  
CRGB leds[NUM_LEDS];
```

```
void setup() {
```

```
    // Uncomment/edit one of the following lines for your  
    leds arrangement.
```

```
    // FastLED.addLeds<TM1803, DATA_PIN, RGB>(leds,  
    NUM_LEDS);
```

```
    // FastLED.addLeds<TM1804, DATA_PIN, RGB>(leds,  
    NUM_LEDS);
```

```
    // FastLED.addLeds<TM1809, DATA_PIN, RGB>(leds,  
    NUM_LEDS);
```

```
    // FastLED.addLeds<WS2811, DATA_PIN, RGB>(leds,  
    NUM_LEDS);
```

```
    // FastLED.addLeds<WS2812, DATA_PIN, RGB>(leds,  
    NUM_LEDS);
```

```
    // FastLED.addLeds<WS2812B, DATA_PIN, RGB>(leds,  
    NUM_LEDS);
```

```
    FastLED.addLeds<NEOPIXEL, DATA_PIN>(leds,  
    NUM_LEDS);
```

```
    // FastLED.addLeds<APA104, DATA_PIN, RGB>(leds,  
    NUM_LEDS);
```

```
    // FastLED.addLeds<UCS1903, DATA_PIN, RGB>(leds,  
    NUM_LEDS);
```

```
    // FastLED.addLeds<UCS1903B, DATA_PIN, RGB>(leds,  
    NUM_LEDS);
```

```
    // FastLED.addLeds<GW6205, DATA_PIN, RGB>(leds,  
    NUM_LEDS);
```

```
    // FastLED.addLeds<GW6205_400, DATA_PIN, RGB>(leds,  
    NUM_LEDS);
```

```
    // FastLED.addLeds<WS2801, RGB>(leds, NUM_LEDS);
```

```
    // FastLED.addLeds<SM16716, RGB>(leds, NUM_LEDS);
```

```
    // FastLED.addLeds<LPD8806, RGB>(leds, NUM_LEDS);
```

```

// FastLED.addLeds<P9813, RGB>(leds, NUM_LEDS);
// FastLED.addLeds<APA102, RGB>(leds, NUM_LEDS);
// FastLED.addLeds<DOTSTAR, RGB>(leds, NUM_LEDS);

// FastLED.addLeds<WS2801, DATA_PIN, CLOCK_PIN,
RGB>(leds, NUM_LEDS);
// FastLED.addLeds<SM16716, DATA_PIN, CLOCK_PIN,
RGB>(leds, NUM_LEDS);
// FastLED.addLeds<LPD8806, DATA_PIN, CLOCK_PIN,
RGB>(leds, NUM_LEDS);
// FastLED.addLeds<P9813, DATA_PIN, CLOCK_PIN,
RGB>(leds, NUM_LEDS);
// FastLED.addLeds<APA102, DATA_PIN, CLOCK_PIN,
RGB>(leds, NUM_LEDS);
// FastLED.addLeds<DOTSTAR, DATA_PIN, CLOCK_PIN,
RGB>(leds, NUM_LEDS);
}

void loop() {
// Turn the LED on, then pause
leds[0] = CRGB::Green;
leds[14] = CRGB::Green;
leds[23] = CRGB::Green;
leds[24] = CRGB::Green;
leds[33] = CRGB::Green;

```

```

leds[1] = CRGB(0,128,0);
leds[15] = CRGB(0,128,0);
leds[22] = CRGB(0,128,0);
leds[25] = CRGB(0,128,0);
leds[32] = CRGB(0,128,0);

FastLED.show();
delay(250);
leds[1] = CRGB::Green;
leds[15] = CRGB::Green;
leds[22] = CRGB::Green;
leds[25] = CRGB::Green;
leds[32] = CRGB::Green;

leds[2] = CRGB(0,128,0);
leds[16] = CRGB(0,128,0);
leds[21] = CRGB(0,128,0);
leds[26] = CRGB(0,128,0);
leds[31] = CRGB(0,128,0);

FastLED.show();
delay(250);
leds[2] = CRGB::Green;
leds[16] = CRGB::Green;
leds[21] = CRGB::Green;
leds[26] = CRGB::Green;

```



```
leds[31] = CRGB::Green;
```

```
leds[3] = CRGB(0,128,0);  
leds[17] = CRGB(0,128,0);  
leds[20] = CRGB(0,128,0);  
leds[27] = CRGB(0,128,0);  
leds[30] = CRGB(0,128,0);
```

```
FastLED.show();  
delay(250);  
leds[3] = CRGB::Green;  
leds[17] = CRGB::Green;  
leds[20] = CRGB::Green;  
leds[27] = CRGB::Green;  
leds[30] = CRGB::Green;  
leds[4] = CRGB(0,128,0);  
leds[18] = CRGB(0,128,0);  
leds[19] = CRGB(0,128,0);  
leds[28] = CRGB(0,128,0);  
leds[29] = CRGB(0,128,0);
```

```
FastLED.show();  
delay(250);  
leds[4] = CRGB::Green;  
leds[17] = CRGB::Green;  
leds[19] = CRGB::Green;
```

```
leds[28] = CRGB::Green;  
leds[29] = CRGB::Green;
```

```
leds[5] = CRGB(0,128,0);  
  
FastLED.show();  
delay(250);  
leds[5] = CRGB::Green;  
leds[6] = CRGB(0,128,0);
```

```
FastLED.show();  
delay(250);  
leds[6] = CRGB::Green;  
leds[7] = CRGB(0,128,0);
```

```
FastLED.show();  
delay(250);  
leds[7] = CRGB::Green;  
leds[8] = CRGB(0,128,0);
```

```
FastLED.show();  
delay(250);  
leds[8] = CRGB::Green;  
leds[9] = CRGB(0,128,0);
```

```
FastLED.show();
```

```
delay(250);  
leds[9] = CRGB::Green;  
leds[10] = CRGB(0,128,0);
```

```
FastLED.show();  
delay(250);  
leds[10] = CRGB::Green;  
leds[11] = CRGB(0,128,0);
```

```
FastLED.show();  
delay(250);  
leds[11] = CRGB::Green;  
leds[12] = CRGB(0,128,0);
```

```
FastLED.show();  
delay(250);  
leds[12] = CRGB::Green;  
leds[13] = CRGB(0,128,0);
```

```
FastLED.show();  
delay(250);  
leds[13] = CRGB::Green;
```

```
FastLED.show();  
delay(250);
```

```
for(int i = 0; i < 33; i++)  
{  
    leds[i] = CRGB::Black;
```

```
}  
FastLED.show();  
delay(250);
```

```
}
```